НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)
“____” _____ 20*18* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю *126 Інформаційні системи та технології*
(код та назва спеціальності)

ОПП *Інформаційні управляючі системи та технології*
(код та назва спеціалізації)

на тему: *Програмно-апаратний комплекс захисту пристроїв окремої підмережі від кібератак з мережі Інтернет*

Виконав : студент *VI* курсу групи ІС-72мп
(шифр групи)

Демиденко Максим Олександрович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *Проф., д.т.н., доц. Стеценко І.В* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *доц. Жданова О.Г.* _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

Факультет *інформатики та обчислювальної техніки*
(повна назва)

Кафедра *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти *другий (магістерський) за освітньо-професійною програмою*

Спеціальність *126 Інформаційні системи та технології*
(код і назва)

ОПП *Інформаційні управляючі системи та технології*
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.А.Павлов
(підпис) (ініціали, прізвище)
«__» *грудня* 20*18* р.

ЗАВДАННЯ

на магістерську дисертацію студенту
Демиденку Максиму Олександровичу
(прізвище, ім'я, по батькові)

1. Тема дисертації *Програмно-апаратний комплекс захисту пристроїв окремої підмережі від кібератак з мережі Інтернет*

науковий керівник дисертації *Д.т.н., доц., проф. Стеценко І.В*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "____" _____ 20*18* р. № _____

2. Строк подання студентом дисертації "*3*" *грудня* 20*18* р.

3. Об'єкт дослідження Процес виявлення шкідливого трафіку пристроїв однієї Підмережі.

4. Перелік завдань, які потрібно розробити *1. Огляд літератури існуючих реалізацій комплексів. 2. Порівняльний аналіз існуючих комплексів. 3. Постановка та формалізація моделі задачі. 4. Створення моделі досліджуємого продукту. 5. Оформлення Документації. 6. Подання роботи на попередній захист. 7. Подання роботи на основний захист.*

5. Орієнтовний перелік ілюстративного матеріалу

1. Структурна схема бази даних. 2. Схема спілкування пристроїв після впровадження програмно-апаратного комплексу. 3. Структурна схема діяльності діяльності визначення типу атаки 4. Структурна схема послідовності визначення типу атаки 5. Структура даних окремого мережевого пакету. 6. Копія екранної форми «Приклад звіту активностей пристрою». 7. Копія екранної форми «Створення правила для розпізнавання активностей»

6. Орієнтовний перелік публікацій

1. *ICCSEEA 2018. The Software and Hardware Complex of Vulnerability Analysis of Devices that are located in One Subnetwork with Access to the Internet.*

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
	Доц. кафедри АСОІУ Жданова О. Г		

8. Дата видачі завдання “ *29* ” *жовтня* 20 *18* р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	*Огляд літератури існуючих реалізацій моделей*	*01.05.2018 – 01.06.2018*	
2	*Порівняльний аналіз існуючих сценаріїв*	*01.06.2018 – 01.07.2018*	
3	*Постановка та формалізація моделі задачі*	*01.07.2018 – 01.09.2018*	
4	*Створення моделі досліджуємого продукту*	*01.09.2018 - 01.11.2018*	
5	*Оформлення документації*	*01.11.2018 – 20.11.2018*	
6	*Подання роботи на попередній захист*	*29.11.2018*	
7	*Подання роботи на основний захист*		

Студент _____ *Демиденко М. О.*
(підпис) (ініціали, прізвище)

Науковий керівник _____ *Стеценко І.В*
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 80 с., 26 рис., 13 табл., 7 додатків, 16 джерел.

Актуальність. На сьогоднішній день питання безпеки інформаційної системи є надзвичайно важливим. Нерідко з'являються повідомлення у засобах масової інформації про те, що новий комп'ютерний вірус став загрозою для нормального функціонування значної частини комп'ютерів. Яскравим прикладом може бути виявлений нещодавно вірус «Wanna Cry», що вражає операційну систему Microsoft Windows шляхом шифрування файлів.

Тому доцільним є створення апаратно-методологічного комплексу, що аналізує і фільтрує трафік в режимі реального часу, формує активності пристроїв, опираючись на пакетні дані.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках ініціативної теми «Методи візуального програмування Петрі-об'єктних моделей» д/р №0117U000918.

Мета дослідження – покращення процесу виявлення загроз з мережі Інтернет, шляхом розробки та впровадження апаратно-методологічного комплексу, що аналізуватиме і фільтруватиме трафік в реальному часі для пошуку шкідливих сигнатур.

Для досягнення мети необхідно виконати наступні **задачі**:

- виконати огляд відомих результатів з розв'язання задачі для пошуку виявлення загроз з мережі Інтернет;
- розробити програмне забезпечення, що буде виявляти шкідливий трафік на основі сигнатур мереж Петрі;
- виконати експериментальне дослідження роботи комплексу
- провести аналіз отриманих результатів.

Об’єкт дослідження – процес виявлення шкідливого трафіку пристроїв однієї підмережі.

Предмет дослідження – методи виявлення шкідливого трафіку пристроїв однієї підмережі та його фільтрацію.

Наукова новизна отриманих результатів

Запропоновано альтернативний архітектурний підхід для впровадження файєрволу в підмережу, шляхом проведення атаки man-in-the-middle. Тобто фізичної взаємодії програмно-апаратного комплексу і пристроїв підмережі немає. Проведено аналіз існуючих підходів до пошуку шкідливого трафіку пристроїв однієї підмережі. Серед проаналізованих методів обраний сигнатурний та поведінковий аналіз. Проведено експерементальне дослідження отриманого комплексу та проаналізовано отримані результати.

МЕРЕЖІ ПЕТРІ, ВРАЗЛИВІСТЬ СИСТЕМИ, АНАЛІЗ ТРАФІКУ, КІБЕР АТАКА

ABSTRACT

Master's dissertation: 80 pages, 26 figures, 13 tables, 7 appendices, 16 sources.

Topicality. Information system security is extremely important today. Often there are messages in the media that the new computer virus has become a threat to the normal functioning of a large part of computers. A striking example may be the recently discovered Wanna Cry virus that affects the Microsoft Windows operating system by encrypting files.

So that, it is expedient to create a hardware and software solution that analyzes and filters traffic in real time, generates activity of devices based on packet data.

Relationship of work with scientific programs, plans, themes. The work was carried out at the Department of Computer-Aided Management and Data Processing Systems (CAMDPS) of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" within initiative theme "Methods of Visual Programming of Petri-Object Models" No. 0117U000918.

The aim of the study – improvement of the process of detecting threats from the Internet, by developing and implementing a hardware and software solution that will analyze and filter real-time traffic to find malicious signatures.

To achieve the goal you need to accomplish the following tasks:

- perform a review of the known results of solving the problem for finding threats from the Internet;
- develop software that will detect malicious traffic based on the signatures;
- to carry out an experimental study of the system functioning;
- to analyze obtained results.

The object of the study is the process of detecting malicious traffic on the devices of one subnet.

Subject of research – methods of detecting malicious traffic of devices of one subnet and its filtration.

Scientific novelty of the obtained results

An alternative architectural approach is proposed for implementing a firewall in the subnet, by running a man-in-the-middle attack. It means that there is no physical interaction of the software and hardware complex and subnet devices. The analysis of existing approaches to finding malicious traffic on devices of one subnet. Among the analyzed methods, a signature and behavioral analysis is selected. An experimental study of the obtained solution was carried out and the obtained results were analyzed.

PETRI NETWORKS, SYSTEM VULNARABILITY, TRAFFIC ANALYSIS, CYBER ATTACK

ЗМІСТ

ВСТУП

На сьогоднішній день питання безпеки інформаційної системи є надзвичайно важливим. Нерідко з’являються повідомлення у засобах масової інформації про те, що новий комп’ютерний вірус став загрозою для нормального функціонування значної частини комп’ютерів. Яскравим прикладом може бути виявлений нещодавно вірус «Wanna Cry», що вражає операційну систему Microsoft Windows шляхом шифрування файлів.

Інформаційна безпека має три основні складові: конфіденційність, цілісність і доступність. Конфіденційність належить до захисту чутливої інформації від несанкціонованого доступу. Цілісність означає захист точності і повноти інформації і програмного забезпечення. Доступність – це забезпечення доступності інформації і основних послуг для користувача в потрібний для нього час.

Враховуючи актуальність інформаційної безпеки було вирішено створити інформаційну систему, що аналізує спираючись на поточний стан обчислювальної системи визначає, які типи атак було проведено на неї.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Об’єкт дослідження: процес виявлення шкідливого трафіку пристроїв однієї підмережі.. Засоби аналізу вразливостей (також відомі як сканери безпеки) представляють собою інструменти управління захистом, які: проводять всебічні перевірки систем, намагаючись локалізувати вразливості захисту, генерують звіт про кількість, природу і силі цих вразливостей, у деяких випадках, як тільки відбувається інцидент, дозволяють дослідникам визначити точку входу і маршрут хакера або порушника. Системи аналізу вразливостей треба доповнювати системами виявлення атак: вони дозволяють системним адміністраторам більш активно захищати свої системи шляхом знаходження і виявлення дірок захисту до того, як хакери зможуть використовувати їх. Системи виявлення атак є за своєю природою реактивними, вони здійснюють контроль за хакерами, націлює на системи, в надії перервати атаки до того, як система буде пошкоджена.

Сервер - у комп'ютерній термінології термін може стосуватися окремого комп'ютера чи програми. Головною ознакою в обох випадках є здатність машини чи програми переважну кількість часу працювати автономно, без втручання людини, реагуючи на зовнішні події відповідно до встановленого програмного забезпечення. Втручання людини відбувається під час встановлення серверу і під час його сервісного обслуговування. Часто це роблять окремі адміністратори серверів з вищою кваліфікацією.

Сервер як програма – програма, що надає деякі послуги іншим програмам (клієнтам). Зв'язок між клієнтом і сервером зазвичай здійснюється за допомогою передачі повідомлень, часто через мережу, і використовує певний протокол для кодування запитів клієнта і відповідей сервера. Серверні програми можуть бути встановлені як на серверному, так і на персональному комп'ютері, щоразу вони забезпечують виконання певних служб (наприклад, сервер баз даних чи веб-сервер).

Комп'ютер або програма, що установлена на цьому комп'ютері, здатні автоматично розподіляти інформацію чи файли під керуванням мережної ОС або у відповідь на запити, надіслані у режимі on-line користувачами, і таким чином надавати послуги іншим комп'ютерам мережі (клієнтам) [1].

Постановка завдання – створити апаратно-методологічний комплекс, що аналізує, виявляє та фільтрує шкідливий трафік пристроїв однієї підмережі з доступом до мережі Інтернет.

Мета даної роботи – покращити процес виявлення загроз з мережі Інтернет, шляхом розробки та впровадження апаратно-методологічного комплексу, що аналізуватиме і фільтруватиме трафік в реальному часі для пошуку шкідливих сигнатур.

Для досягнення мети необхідно виконати наступні **задачі**:

- виконати огляд відомих результатів з розв’язання задачі для пошуку виявлення загроз з мережі Інтернет;
- розробити програмне забезпечення, що буде виявляти шкідливий трафік на основі сигнатур мереж Петрі;
- виконати експериментальне дослідження роботи комплексу
- провести аналіз отриманих результатів

Опис процесу діяльності

Розглянемо дії, які має виконати користувач для повномірного користування системою за допомогою UML діаграми діяльності. Діаграма діяльності для визначення типу атаки зображена на рисунку 1.1.

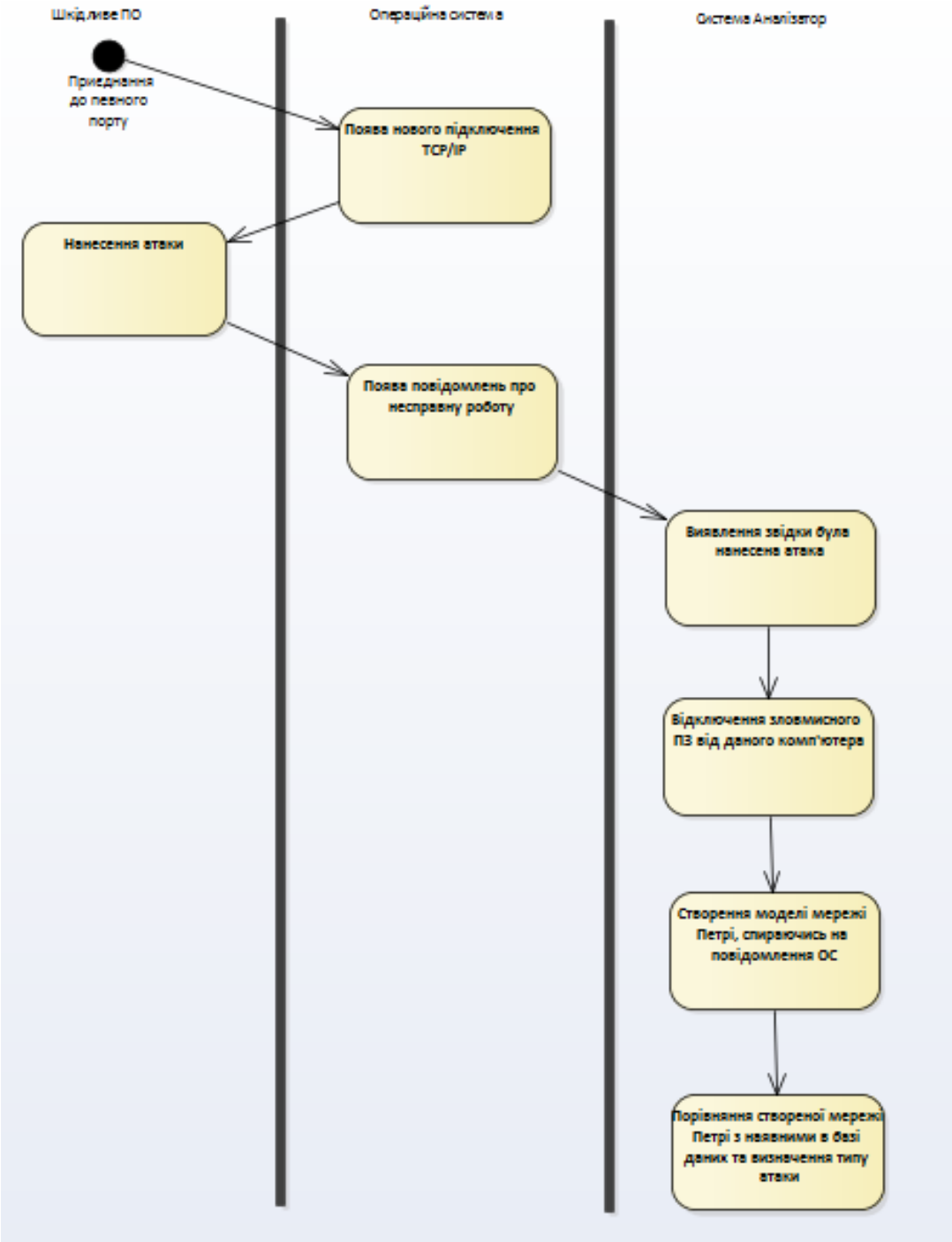


Рисунок 1.1 – Діаграма діяльності бізнес-процесу Визначення типу атаки

Опис функціональної моделі

Для проектування діаграми використання спочатку необхідно визначити дійових осіб (акторів), а потім визначити, які дії у системі може виконувати кожен з акторів. Список акторів системи:

- апаратно- методологічний комплекс.

Структурна схема варіантів використання представлена на схемі структурній варіантів використання у графічному додатку.

У таблиці 1.1 розглянуто детальніше варіанти використання:

Таблиця 1.1 – Формулювання функціональних вимог до варіантів використання

Актор	**Варіант використання**	**Функціональна вимога**	**Пріоритет**
Апаратно-методоло-гічний комплекс	Використання бібліотеки Nmap для сканування мережі	1 Комплекс виконує пошук доступних приладів для здійснення атаки, які дозволяють виявити та вивести дані на екран додатка Nmap 1.1 Виконується підключення до мережі (wi-fi, lan и т.д.), сканування і відображення усіх підключених приладів. 1.1.1 Визначення ролей усіх кінцевих невідомих приладів у мережі	Середній Середній Середній
	Проведення атаки man-in-the-middle	2. Для початку перехвату пакетів виконується атака man-in-the-middle для можливості перехопити пакети певного пристрою. 2.1 При передачі пакетів від пристроя до роутера «вдавати» роутер для пристрою 2.2 При передачі пакетів від роутера до пристроя – «вдавати» пристрій для роутера	Середній

Продовження таблиці 1.1

	Аналіз трафіку	3. Перевірка трафіку на сигнатури 3.1 Виявлення загроз за довомогою сигнатур змодельованих на мержах Петрі. 3.2 Блокування шкідливих пакетів даних	Високий Високий Високий
	Формування активностей пристрою	4. Трансформація пакетів даних в активності. 4.1 Композиція пакетів за параметрами у оригінальні одиниці – флоу. 4.2 Визначення сервісу флоу за DNS Request 4.3 Визначення сервісу флоу за NDPI protocol 4.4 Визначення сервісу флоу за портами	Середній Середній Середній Середній Середній

1.2 Огляд наявних аналогів

В ході пошуку схожих за функціональністю систем було виявлено дві системи зі схожими функціями, які на даний момент є у продажу:

- система Qualys ThreatPROTECT;
- система BeyondTrust Retina CS.

Основними функціями системи Qualys ThreatPROTECT є:

- підтримка в курсі останніх аналізів вразливостей та оголошенньь;
- завдяки потужнім можливостям кореляції показує, на скільки ваші IT-ресурси схильні до кожного розкриття інформації;
- дозволяє розгорнути і отримати докладну інформацію конкретних вразливостей та вразливих IT-ресурсів;

- дозволяє тонко налаштовувати спиок елементів шляхом фільтрації і сортування елементів у відповідності з різними критеріями;
- відображає всю картину загрозу з першого погляду;
- забезпечує динамічно налаштовуваємі подання з визначеною статистикою;
- дозволяє подивитися і отримати доступ до додаткової інформації про активи, помічених як вразливі;
- дає потужний інструмент для постійного пошуку конкретних активів і вразливостей;
- дозволяє створювати запити ad-hoc з декількома змінними і критеріями, таких як: клас активів, тип вразливості і операційної системи;
- дозволяє додатково сортувати результати пошуку, фільтрувати та уточнювати;
- дозволяє зберігати запити і перетворювати їх на постійні представлення приладової панелі.

Основними функціями системи BeyondTrust Retina CS є:

- знаходження мереж, Інтернет, мобільних, хмаровіх, віртуальних і інфраструктур IoT;
- профіль конфігурації активів і потенційних ризиків;
- точкові вразливості, шкідливи програми та атаки;
- аналіз потенціальних загроз і повернення на реабілітацію;
- усунення вразливостей через інтегроване управління виправленнями;
- звіт про вразливість, дотримання, контрольни точки і т.д.;
- захист кінцевих точок проти атак на стороні клієнта.

Як бачимо, обидві програми в принципі мають схожі функції, і виконують основну задачу – пошук та аналіз вразливостей.

1.3 Постановка задачі

Призначення розробки

Система аналізу вразливостей обчислювальної системи призначена для інформаційно-аналітичного забезпечення користувачів у частині виконання таких процесів:

- визначення підключень до операційної системи;
- виявлення місця нанесення атаки на операційну систему;
- відключення шкідливого програмного забезпечення від ОС;
- визначення типу нанесеної атаки.

Цілі та задачі розробки

Основними цілями розробки веб застосування є:

- підвищення безпеки обчислювальної системи.

Для досягнення поставлених цілей мають бути вирішені такі задачі:

- отримання інформації про наявні пристрої в підмережі;
- виконати моніторинг трафіку пристрою;
- перевірити сигнатури;
- сформувати список активностей.

Висновоки до розділу

Виконано проектування системи аналізу вразливостей обчислювальної системи: проаналізовано предметне середовище; розглянуто процес діяльності, відповідно до якого були виділені групи користувачів, що взаємодіють з системою, та визначені їх функції; розроблена діаграма діяльності, яка визначає порядок виконання дій акторів при роботі з порталом; визначені призначення і цілі розробки та задачі, які необхідно реалізувати.

2 МЕТОДИ ТА МОДЕЛІ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ АНАЛІЗУ ВРАЗЛИВОСТЕЙ ПРИСТРОЇВ ОКРЕМОЇ ПІДМЕРЕЖІ

2.1 Вхідні дані

Вхідні дані вводяться в систему користувачем за допомогою EMDS (Extendable malvare detection system) – міні-компьютер на базі ОС Linux з встановленими прогамами для сканування та аналізу вразливостей).

Дані від програми Nmap:

- перелік усіх пристроїв що присутні у мережі;
- IP-адрес кожного пристрію;
- назва та призначення кожного пристрою (визначення ролі пристрою у мережі);
- ОС що працює на даному пристрії;

Дані від програми Tshark:

- перелік пакетів у форматі JSON.

2.2 Вихідні дані

Перелік вихідних документів:

- звіт з активностями по кожному пристрою підмережі;
- загальний звіт по всім пристроям підмережі;
- вивід на екран повідомлення про здійснення атаки на систему та вид цієї атаки.

2.3 Математичне забезпечення

Загальні вимоги

Розпізнавання комп'ютерних атак в динаміці функціонування ІС являє собою аналіз і виявлення тих параметрів, які характеризують дію атаки. Для опису станів інформаційної системи в повній мірі підходять мережі Петрі та імітаційне моделювання.

Під атакою на ІС розуміють дії або послідовність зв'язаних між собою дій порушника, які приводять до реалізації загроз інформаційним ресурсам, шляхом використання вразливостей цієї ІС.

Множину можливих типів комп'ютерних атак на ІС представимо у вигляді:

$\Omega = \{\omega_1,...,\omega_k\}$,

де Ω – множина типів комп'ютерних атак;

ω_i – тип комп'ютерних атак.

Множину вразливостей ІС представимо у вигляді:

$\Phi = \{\varphi_1,..., \varphi_k\}$,

де Φ – множина вразливостей;

φ_i – вразливість ІС.

Математична постановка задачі

Призначенням цієї задачі є здійснення атаки на інформаційну систему, сканування системи на вразливості, а також визначення типу атаки за допомогою моделі поведінки системи мережі Петрі.

Складемо перелік загроз і вразливостей, які потенційно можуть існувати в системі із заданим набором функціональних можливостей.

Дано:

множина загроз T

множина вразливостей V.

Множина загроз T може бути описано наступним чином:

$T = (I, V, S_t)$,

де:

I - рівень порушника, який може реалізувати загрозу;

V - множина вразливостей, експлуатація яких приводить до успішної реалізації загрози;

S_t - множина векторів стану, що описують неприпустимий стан, в яке перейде система при успішній реалізації загрози в момент часу t.

Множина вразливостей V може бути описано наступним чином:

$V = (R, P_t)$,

де:

- R - множина рекомендацій щодо усунення вразливостей;
- P_t - множина станів системи, в яких вразливість може існувати.

Визначити:

Множину успішних атак проведених на ІС.

Обґрунтування методу розв'язання

Функціональні можливості моделі ІС можуть бути описані в термінах теорії мережі Петрі, тому що таке уявлення дозволяє найбільш детально і наочно описати функціонування системи і надалі вказати в яких станах ІС потенційно можуть існувати уразливості.

Визначення. Мережа Петрі є четвіркою, $C = (P, T, I, O)$. $P = \{p_1, p_2, ..., p_n\}$ - кінцеве безліч позицій, $n \geq 0$. $T = \{t_1, t_2, ..., t_n\}$ - кінцева множина переходів, $m \geq 0$. Множина позицій і множина переходів не перетинаються, $P \cap T = \varnothing$.

$I: T \rightarrow P^{\infty}$ є вхідний функцією - відображенням з переходів в комплекти позицій.

$O: T \rightarrow P^{\infty}$ - вихідна функція - відображення з переходів в комплекти позицій.

Просте уявлення системи мережею Петрі засновано на двох основоположних поняттях: події та умови. Події - це дії, які відбуваються в системі. Виникненням подій управляє стан системи. Стан системи може бути описано множиною умов.

Для визначення стану, в яке переходить ІС в момент спрацьовування переходу t, будемо використовувати набір параметрів:

- c_t - порт закритий;

- i_t - порт відкріт;

такі, що $c_t, i_t \in \{0,1\}$.

Причому $c_t = 1$ (аналогічно $i_t = 1$), якщо в даному стані забезпечується конфіденційність (відповідно цілісність, доступність) інформації, що обробляється в ІС. У разі якщо порушується конфіденційність і/або цілісність і/або доступність, параметри приймають нульове значення $c_t = 0$, відповідно $i_t = 0$. Таким чином, стан системи в момент спрацьовування переходу може бути представлено $s_t = (c_t, i_t) \in S$.

Стан системи, в якому вектор $s_t = (1,1)$ будемо називати безпечним станом (S^+). Стан, в якому хоча б один з параметрів дорівнює нулю - небезпечним чи недопустимим станом (S^-). $S^+ \cup S^- = S$,

де S - множина всіх можливих станів системи.

Опис методів розв'язання

Апріорний метод протидії комп'ютерним атакам на інформаційну систему призначений для формалізації процесів компенсації впливів атак, реорганізації інформаційно-обчислювального процесу, коригування регламентів виконання розрахункових програм, вироблення керуючих впливів і відновлення стійкості функціонування ІС.

Найбільшою мірою досягненню цілей розробки достовірної та адекватної математичної моделі опису процесів протидії комп'ютерним атакам в порівнянні з іншим математичним апаратом відповідають характеристики мереж Петрі.

Графічне представлення процесів роботи ІС спільно з процесами протидії атакам за допомогою елементів мереж Петрі зручно для дослідження та інтерпретації, легко і просто перетворюється в моделюючі алгоритми і програми натурного і імітаційного моделювання. При апріорному оцінюванні характеристик ІС їх опис за допомогою мереж Петрі дозволяє уявити вигляд структури системи, визначити механізм впливу атак, виділити групи інформаційних і керуючих потоків.

Для формалізації інформаційно-обчислювального процесу в ІС при впливі комп'ютерних атак і наступних процедур його відновлення використовуємо мережу Петрі (МП).

На основі аналізу підходів до формального подання мережі Петрі пропонується апріорний метод протидії комп'ютерним атакам на ІС в термінах МП визначити у вигляді набору типових математичних елементів:

$S_{IC} = <(P, V), T, D, M, Q, Ip, Y>$,

де $P = p_1, p_2, ..., p_i$ - непорожня скінченна множина позицій, що характеризують штатний режим функціонування ІС.

$V = v_1, v_2, ..., v_j$ - множина позицій відновлення, що відбивають процедури відновлення при впливі комп'ютерних атак $T = t_1, t_2, ..., t_n$ - непорожня скінченна множина переходів.

D - непорожня скінченна множина дуг мережі, причому

$D = (D_1 \cup D_2)$

$D_1 = (P \times T) \cup (V \times T)$ - непорожня множина вхідних дуг, що з'єднують позиції і переходи,

$D_2 = (T \times P) \cup (T \times V)$ - непорожня множина вихідних дуг, орієнтованих від переходів до позицій;

M - множина маркувань позицій мережі Петрі;

$F_p: (M_p: P \rightarrow N), F_v: (M_v: V \rightarrow N)$ - функції початкового маркування позицій штатного функціонування і відновлення відповідно, N = {0,1,2, ...} - множина натуральних чисел;

Q - множина ймовірностей запусків переходів, що відображає ймовірність знаходження ІС в режимі штатного функціонування, в моменти впливу комп'ютерних атак або в процесі відновлення;

$I = i_{p1}, i_{p2}, ..., i_{pm}$ - множина пріоритетів для дуг;

$Y = y_1, y_2, ..., y_k$ - множина часових параметрів комп'ютерних атак, що визначає час спрацьовування переходу що моделюють виявлення комп'ютерних атак і реакцію ІС на них.

При цьому під позиціями мережі розуміються реальні процеси в системі, а переходи дозволяють оцінити часові характеристики процесів протидії атакам.

Запропонований апарат мереж Петрі орієнтований на подієво-стохастический підхід до узагальненого аналізу впливу атак на ІС, який полягає в поданні інформаційно-обчислювальних процесів у вигляді подій (інтерпретуються переходами МП), змінюющихся станів (набір множин *P*, *V*, *T*, *M*) і випадкових процесів, формалізованих набором множин *(Q, I_p, Y)* і відображають імовірнісні умови переходу ІС з одного стану в інший.

Позиції мережі Петрі інтерпретуються умовами необхідними для здійснення того чи іншого процесу. В мережі Петрі введені множини позицій відновлення *V*, ймовірностей запусків переходів *Q*, пріоритетів для дуг I_p, параметрів комп'ютерних атак *Y*, що забезпечують формалізацію процесу функціонування ІС при впливі комп'ютерних атак. Елементи мережі Петрі є основою для моделювання процесів функціонування ІС, умов виникнення комп'ютерних атак і процесів протидії їм.

Апріорний метод протидії комп'ютерним атакам в термінах МП формалізує процес функціонування реальної ІС і подій протидії комп'ютерним атакам в структурно-параметричному вигляді: структури процесів функціонування ІС в умовах впливу комп'ютерних атак - моделлю графів, а параметри ІС, комп'ютерних атак і засобів протидії атакам - математичними термінами МП.

Даний метод представляється у вигляді двух елементів:

1. Моделі реалізації комп'ютерних атак на ІС в термінах мереж Петрі, що формалізує механізми реалізації комп'ютерної атаки на ІС і порядок подолання рубежів протидії відповідно до плану порушника

2. Моделі протидії комп'ютерним атакам на ІС в термінах мереж Петрі, що представляє собою типовий граф і математичні співвідношення для опису процесів попередження, виявлення, аналізу комп'ютерних атак і активної протидії атакам.

Модель реалізації процесу впливу комп'ютерних атак на ІС в термінах

МП призначена для детальної формалізації процедур впливу атак на ІС. Мінімальний набір, необхідний для відображення процесів, пов'язаних з реалізацією атак на ІС, виражається співвідношенням:

$$\exists\, min(V,Q,D_z,I_p,Y\ ) = S',\ \ \phi : S' \rightarrow S\ ,$$

де V – множина позицій відновлення, що відображають процедури відновлення після атаки;

Q – ймовірність запуску переходів відображаюча ймовірність нормальної роботи системи;

I – множина пріоритетів для дуг;

Y – множина часових параметрів комп'ютерних.

Аналіз послідовності елементів множин з мінімального набору S^* проводиться на основі подання цих елементів наступними функціями:

$$\forall(v_j\ ,q_w,d_{zc}\ ,i_{pm}\ ,\ y_k\ ) \rightarrow \exists\psi_1 : \{\ 1,2,\dots\ j\ \} \rightarrow \{v_1,v_2\ ,\dots v_{\ j}\},$$

$$\exists\psi_2 : \{\ 1,2,\dots w\ \} \rightarrow \{q_1,q_2,\dots q_w\},$$

$$\exists\psi_3 : \{\ 1,2,\dots n\ \} \rightarrow \{d_{z1},d_{z2}\ ,\dots d_{zn}\ \},$$

$$\exists\psi_4 : \{\ 1,2,\dots i\ \} \rightarrow \{i_{p1},i_{p2},\dots i_{pm}\ \},$$

$$\exists\psi_5 : \{\ 1,2,\dots k\ \} \rightarrow \{y_1,\ y_2\ ,\dots y_k\}.$$

Введемо L-множину індексів, тоді для кожного $l \in L$ існує S_l, якою є підмножина множини S^*. Множина$\{(S'_l)\ l \in L\}$ є сімейством підмножин множини S^*. Функція $\psi: L \rightarrow S'$, значеннями якої є множини, являє собою сімейство множин. Для цього сімейства множин можна записати співвідношення $\psi(\ l) = S'_l$ інакше позначити його $(S'_l)\ l \in L$. Вступ сімейства множин $(S'_l)\ l \in L$ дозволяє визначити функцію вибору через об'єднання сімейства підмножин в наступному вигляді:

$$\exists\,\psi : L \rightarrow \bigcup_L S'_l, \forall l(l \in L \rightarrow (l) \in S_l)$$

Графічним представленням мережі Петрі є орієнтований мультіграф з вершинами трьох типів з множин $(P,\ V,\ T)$ і дугами з множин D_1, D_2. Граф дозволяє в статичному вигляді задати структуру ІС.

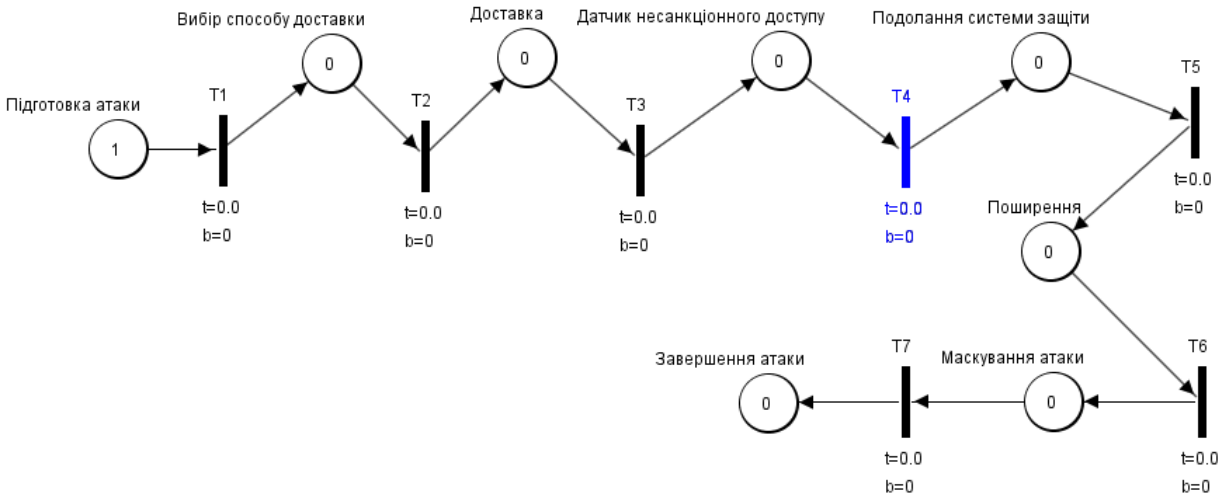


Рисунок 2.1 - Модель реалізації комп'ютерних атак на ІС в термінах мереж Петрі

Модель протидії комп'ютерним атакам на ІС в термінах мереж Петрі базується на моделі динамічних процесів протидії комп'ютерним атакам. Ця модель поєднує в своєму складі співвідношення, що формалізують процеси збору, обробки, зберігання, передачі та відображення інформації в ІС при виконанні технологічного циклу управління, тип ПО, топологію мережі передачі даних, можливі сценарії атак і реакцію засобів протидії на них. Стратегія моделювання процесів протидії комп'ютерним атакам в термінах МП здійснюється відповідно до класифікації комп'ютерних атак.

Для математичного опису в термінах мереж Петрі комплексного вирішення проблеми протидії комп'ютерним атакам на ІС в загальному випадку необхідно:

- визначити даний стійкий стан функціонування ІС (P_{i+1}, n_{i+1}, e_{i+1}, b_{i+1}) в умовах впливу атак на основі даних про минуле стійкому стані ІС (P_i, n_i, e_i, b_i) до впливу атак (вихідний стан ІС до впливу атаки);
- спрогнозувати майбутній стійкий стан ІС (P_{i+2}, n_{i+2}, e_{i+2}, b_{i+2}) на основі наявного досвіду протидії комп'ютерним атакам

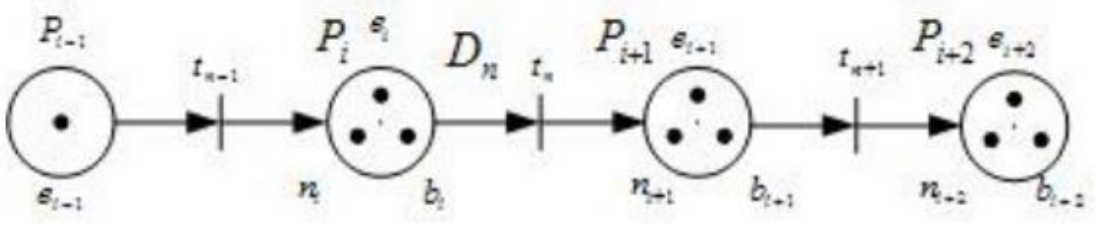


Рисунок 2.2 - Схема можливих станів життєвого циклу ІС в термінах мереж Петрі

На рисунку 2.2 прийняті наступні позначення:

n_i - параметр, що характеризує даний стійкий стан ІС на *i*-й момент часу;

e_i - параметр, що описує минуле стійкий стан ІС на *i*-й момент часу;

b_i - параметр, який представляє майбутній стійкий стан ІС на *i*-й момент часу;

P_i - непорожня скінченна множина позицій;

n_t - непорожня скінченна множина переходів;

D_n - множина дуг мережі;

$e_i \in E_{ki}$, $n_i \in N_{ki}$, $b_i \in B_{ki}$ - обмеження на МП: нижній індекс «к» відображає, що існують кінцеві множини значень параметрів про минуле, сьогодення і майбутнє у стійкому стані ІС.

Динаміка роботи мережі Петрі зображена на рисунку 3.2, де визначається F_p: $(M_p: P \rightarrow M)$ - функцією початкового маркування. Маркування з трьох точок в позиціях P_i, P_{i+1}, P_{i+2} означає, що для ефективної протидії комп'ютерним атакам в *i*-й момент часу необхідно визначити параметри справжнього, минулого і майбутнього стійкого стану ІС на основі моделювання динамічних процесів функціонування системи.

2.4 Опис моделі перехоплення трафіку

Брандмауери, проксі-сервери, демілітаризовані зони (DMZ) - компанії все більше розгортають таку тактику, щоб захистити свої приватні мережі від небезпек Інтернет. Але не всі напади виходять ззовні. Найслабкішою ланкою в ланцюгу мережевої безпеки є локальна мережа (LAN). Зловмисник, який вже знаходиться в мережі, має безліч доступних способів перегляду трафіку даних і керування ними за бажанням. Внутрішні зловмисники використовують вразливість протоколів ARP. Це використовується з мережами Ethernet на основі IPv4 для вирішення IP-адрес на MAC-адреси, що представляє адміністраторам проблеми безпеки.

Записами ARP можна легко керувати, використовуючи фальшиві пакети даних. Ці випадки згадуються з використанням терміна ARP spoofing, атака між людьми, яка дозволяє хакерам перейти непоміченими між двома системами зв'язку. Тут ми показуємо, як вирішення адреси може бути спеціально оброблено через ARP і запропонувати можливі контрзаходи.

ARP-spoofing (також відомий як ARP-poisoning) реалізує атаки "man-in-the-middle", що виконуються в таблицях ARP локальної мережі. Ця форма нападу призводить до того, що хакери надсилають підроблені пакети ARP, які слайд між двома системами зв'язку, непоміченими, щоб вони могли слухати або керувати своїм трафіком даних.

На відміну від пристроїв в Інтернеті, пристрої в локальній мережі не спілкуються безпосередньо через IP-адреси. Замість цього вони використовують фізичні адреси апаратури для адресації в локальних мережах IPv4. Ці MAC-адреси (Media Access Control) є унікальними 48-бітовими цифрами і дають змогу ідентифікувати кожен пристрій в локальній мережі через мережеву карту.

Наприклад MAC-адреса: 00-80-41-ae-fd-7e

MAC-адреси призначаються їх відповідними виробниками апаратного забезпечення та унікальні у всьому світі. Теоретично ці апаратні адреси будуть придатними для глобальної адресації. Але на практиці це не спрацьовує, оскільки

адреси IPv4 занадто короткі, щоб повністю відобразити MAC-адресу. У мережах, що базуються на IPv4, вирішення адреси через ARP неминуче.

Якщо комп'ютер A хоче зв'язатися з комп'ютером B в тій же мережі, спочатку він повинен визначити відповідну MAC-адресу для своєї IP-адреси. Це використовує протокол розпізнавання адрес (ARP), мережевий протокол, який працює відповідно до схеми відповідей запиту.

Після пошуку відповідної MAC-адреси комп'ютер A надсилає запит на трансляцію (або запит ARP) на всі пристрої в мережі. Цей запит містить таку інформацію:

Комп'ютер з MAC-адресою xx-xx-xx-xx-xx-xx та IP-адресою yyy.yyy.yyy.yyy хотів би зв'язатися з комп'ютером з IP-адресою zzz.zzz.zzz.zzz і вимагає відповідну MAC-адресу.

Запит ARP приймається всіма комп'ютерами в локальній мережі. Щоб запобігти передачі запиту ARP перед відправкою кожного пакета даних, кожен комп'ютер у мережі виконує локальний набір, який називається кеш ARP. У цих таблицях всі відомі MAC-адреси тимчасово зберігаються поряд з їх відповідними IP-адресами.

Таким чином, всі комп'ютери в мережі записують запит на трансляцію разом із супровідною адресою відправника. Відповідь на запит на трансляцію очікується лише від комп'ютера B. Його відповідь ARP містить таку інформацію:

Це система з IP-адресою zzz.zzz.zzz.zzz. Запитана MAC-адреса є aa-aa-aa-aa-aa-aa.

Якщо ця ARP-відповідь доставляється до комп'ютера A, то вона містить всю інформацію, необхідну для передачі пакетів даних на комп'ютер B. Комунікації через локальну мережу тепер нічим не перешкоджають.

Якщо призначений комп'ютер не відповідає, але замість цього відповідь надходить з іншого пристрою, який контролюється внутрішнім атакуючим із кримінальними намірами - саме тут відбувається підроблення ARP.

Схема відповіді запитів протоколів ARP організована таким чином, що перша відповідь на запит ARP приймається і зберігається. У контексті підробки ARP хакери намагаються вигнати фактичний цільовий комп'ютер, щоб

відправити відповідний пакет з невірною інформацією та керувати таблицею ARP запитуючого комп'ютера. Це називається отруєнням ARP або "забрудненням" кеш-пам'яті ARP. Як правило, ці пакети даних містять MAC-адресу мережного пристрою, що контролюється хакерами. Цільова система потім зв'язує вихідний IP з неправильною адресною адресою та надсилає всі майбутні пакети даних до хакерської системи. Ця система тепер має можливість записувати або керувати всіма трафіками даних.

Щоб залишатися невиявленим, трафік перехоплених даних зазвичай передається до фактичної цільової системи. Потім хакер стає людиною посередині. Якщо перехоплені пакети даних не пересилаються, але замість цього їх відкидають, підроблення ARP може призвести до відмови в обслуговуванні (DoS). ARP-spoofing функціонує як у локальних мережах, так і в мережах WLAN. Навіть шифрування бездротових мереж за допомогою Wi-Fi Protected Access (WPA) не забезпечує захисту. Для комунікації в локальних мережах IPv4 всі підключені пристрої повинні вирішити MAC-адреси - це можна зробити лише через ARP.

Одне добре відоме програмне забезпечення, яке спеціально закріплюється на запитах на трансляцію та відповідає підробленими ARP-відповідями, є Cain & Abel. Але щоб "заразити" ARP-кеш-пам'ять мережевого пристрою, хакеру не обов'язково треба чекати на запити ARP. Інша стратегія включає постійне бомбардування мережі з помилковими відповідями ARP. Хоча більшість систем ігнорують пакети відповідей, які не можна призначити запиту, це змінюється, як тільки комп'ютер у локальній мережі запускає запит ARP, і тому готовий отримати відповідь. Залежно від часу, перша відповідь цільової системи або один з підроблених пакетів відповідей надійде до відправника. Цей тип атаки може бути автоматизований такими програмами, як Ettercap.

Якщо хакер успішно перемикається між двома партнерами по зв'язку, вони можуть вільно перебувати незахищених з'єднань. Оскільки вся передача зламаного з'єднання проходить через систему хакера, вона може читати та керувати даними за бажанням. Захист від шпіонажу даних може обіцятись за допомогою деяких методів шифрування та сертифікатів для автентифікації.

Якщо зловмисник захоплює лише закодовані дані, найгірший випадок обмежується лише відмовам у наданні послуг, відкидаючи пакети даних. Але надійне шифрування даних має здійснюватися послідовно.

Численні інструменти, які можуть використовуватися для атак "людина в середині", забезпечують функції підробки ARP, а також реалізацію клієнта та сервера для SSL / TLS, SSH та інших протоколів шифрування. Вони мають можливість наслідувати відповідні сертифікати та встановлювати зашифровані з'єднання. Наприклад, Cain & Abel імітує веб-сервер із підтримкою SSL, який потім надсилає до системи жертви ненадійний SSL-сертифікат. Слід зауважити, що в цьому випадку користувачі мережі попереджають, але ці застереження, як правило, або ігноруються або неправильно інтерпретуються користувачем, тому уроки з питань безпеки мережі також повинні охоплювати відповідальне поводження з цифровими сертифікатами.

Оскільки ARP-spoofing використовує протокол вирішення адреси, всі мережі IPv4 схильні до атак такого роду. Реалізація IPv6 також не могла вирішити цю основну проблему. Новий IP-стандарт відмовляється від ARP, а замість цього контролює вирішення адреси в локальній мережі за допомогою протоколу NDP (Neighbor Discovery Protocol), який також є вразливим для підробки атак. Безпека може бути закрита через протокол Secure Neighbor Discovery (SEND), але це не підтримується багатьма настільними операційними системами.

Можливим захистом від маніпулювання ARP кешами пропонують статичні записи ARP, які можна встановити в Windows, наприклад, за допомогою програми командного рядка ARP та команди arp -s. Але оскільки записи цього типу мають бути зроблені вручну, ці методи безпеки, як правило, обмежуються лише найважливішими системами в мережі.

Ще однією мірою проти зловживань ARP є поділ мереж на перемикачі рівня 3. Неконтрольовані запити трансляції охоплюють лише ті системи, що знаходяться в одному сегменті мережі. ARP-запити в інших сегментах перевіряються перемикачем. Якщо вони працюють на рівні мережі (Layer 3), тоді IP-адреса збігається як з MAC-адресою, так і з попередніми записами. Якщо є

будь-які розбіжності або часті перепризначення, звучить сигнал перемикача. Але необхідне обладнання досить дороге. Адміністратори повинні визначити, чи підсилює рівень безпеки, виправдовує фінансові витрати. З іншого боку, значно більш сприятливі перемикачі Layer 2, які працюють на рівні каналу передачі даних, не є адекватними. Хоча вони реєструють зміну MAC-адреси, призначення до відповідної IP-адреси залишається незмінним.

2.5 Опис структури бази даних

Первинний ключ в таблиці 2.1 позначений за допомогою підкреслювання. В кожної сутності наявний атрибут id, який є первинним ключем, якщо немає іншого первинного ключа.

Таблиця 2.1 – Опис структури бази даних

Назва таблиці	**Назва атрибута**	**Таблиця, на яку посилається**	**Тип даних**	**Детальна інформація**
User – зберігає користувачів	id		string	
	email		string	Електронна пошта користувача
	password		string	Зашифрований пароль користувача
	fname		string	Ім’я користувача
	lname		string	Прізвище користувача
	phone		string	Номер телефону

Продовження таблиці 2.1

	createdAt		datetime	Дата створення
	updatedAt		datetime	Дата редагування
	deletedAt		datetime	Дата видалення
Settings – зберігає налаштування програмно-апаратного комплексу	id		string	Ключ дозволу
	tz		string	Ключ ролі
	arp		string	Дата створення
	name		string	Дата редагування
	wfmId		string	Дата видалення
	dailyReport		datetime	Час відправлення щоденного звіту
	lanInterface		string	LAN-інтерфейс
	lanGateway		string	LAN-шляхопровід
	emails		string	Емейли, куди надсилати листи зі звітами
	version		string	Версія прошивки
	isInspectPackets		boolean	Флаг аналізу пакетів
	isNewDeviceAlert		boolean	Флаг сповіщення при появі нового пристроя в мережі
	isMonitorAll		boolean	Флаг моніторингу активностей всіх пристроїв
	isSetup		boolean	Флаг проходження авторизації
	createdAt		datetime	Дата створення
	updatedAt		datetime	Дата редагування

Продовження таблиці 2.1

	deletedAt		datetime	Дата видалення
Packet – зберігає пакети	id		string	Ключ дозволу
	macSrc	User	integer	macSrc
	macDst		string	macDst
	ipSrc		string	ipSrc
	ipDst		string	ipDst
	dnsResponceNames		string	dnsResponceNames
	dnsResponceName		string	dnsResponceName
	dnsResponceIps		string	dnsResponceIps
	dnsResponceIp		string	dnsResponceIp
	dnsRequests		string	dnsRequests
	dnsRequest		string	dnsRequest
	ndpiProtocol		string	ndpiProtocol
	dnsId		string	dnsId
	frameLen		string	frameLen
	ports		string	Порти
	createdAt		datetime	Дата створення
	updatedAt		datetime	Дата редагування
	deletedAt		datetime	Дата видалення
	flowId	Flow	string	Ідентифікатор Flow
Flow – зберігає дані об’єднаних пакетів	id		varchar	Ключ атаки
	activityId	Activity	varchar	Значення атаки
	segments		timestamp	Дата створення
	title		timestamp	Дата редагування
	statisticType		timestamp	Дата видалення
	dnsResponceNames		string	dnsResponceNames
	dnsResponceName		string	dnsResponceName
	dnsResponceIps		string	dnsResponceIps
	dnsResponceIp		string	dnsResponceIp

Продовження таблиці 2.1

	dnsRequests		string	dnsRequests
	dnsRequest		string	dnsRequest
	ndpiProtocol		string	ndpiProtocol
	dnsId		string	dnsId
	dnsRCode		string	dnsRCode
	receivedPackets		integer	Кількість отриманих пакетів
	sentPackets		integer	Кількість відправлених пакетів
	receivedBytes		integer	Кількість отриманих байт
	sentBytes		integer	Кількість відправлених байт
	bandwidth		integer	Загальна сума байт
	bandwidthRatio		float	Загальна сума байт поділена на
	density		float	Щільність трафіку
	differenceTime		float	Різниця між часом початку та часом кінця у секундах
	source		string	source
	ports		string	Порти
	port		string	Порт
	startTime		datetime	Час початку
	endTime		datetime	Час кінця
	updatedAt		datetime	Дата створення
	createdAt		datetime	Дата редагування

Продовження таблиці 2.1

Device – зберігає дані про пристрої підмережі	id		string	Ідентифікатор пристрою
	ip		string	IPv4-адреса
	ipv6		string	IPv6-адреса
	key		string	Дата створення
	mac		string	MAC-адреса
	os		string	Операційна система пристрою
	os_version		string	Версія операційної сисеми
	name		string	Ім'я пристрою
	type		string	Тип
	vendor		string	Вендор
	netbios		string	netbios
	hostname		string	hostname
	description		string	Опис
	imageURL		string	URL зображення
	icon		string	Іконка
	isOnline		boolean	Флаг чи онлайн
	isMonitor		boolean	Флаг чи увімкнений моніторинг
	isPause		boolean	Флаг чи пристрій на паузі
	isSetIcon		boolean	Флаг чи встановлена іконка

Продовження таблиці 2.1

	isSetName		boolean	Флаг чи встановлена назва
	isSetHostname		boolean	Флаг чи встановлено hostname
	pauseStartedAt		timestamp	Час початку обмеження інтернету
	monitorAt		timestamp	Час моніторингу
	updated_at		timestamp	Дата редагування
	created_at		timestamp	Дата створення
Activity	id		string	Ідентифікатор активності
	deviceId	Device	string	Ідентифікатор пристрою
	statisticType		string	Категорія
	service		string	Сервіс
	sentBytes		integer	Кількість відправлених байт
	sentPackets		integer	Кількість відправлених пакетів
	receivedBytes		integer	Кількість отриманих байт
	receivedPackets		integer	Кількість отриманих пакетів

Продовження таблиці 2.1

	differenceTime		integer	Різниця між часом початку та часом кінця
	upArrow		boolean	Флаг вхідна чи вихідна активність
	inStatistic		boolean	Флаг включення в статистику
	startTime		datetime	Час початку
	endTime		datetime	Час кінця
	statisticId	Statistic	string	Ідентифікатор статистики
	updated_at		timestamp	Дата редагування
	created_at		timestamp	Дата створення
Statistic	id		string	Ідентифікатор статистики
	deviceId	Device	string	Ідентифікатор пристрою
	key		string	Ключ статистики
	hour		string	Година
	type		string	Категорія
	sentBytes		integer	Кількість відправлених байт
	sentPackts		integer	Кількість відправлених пакетів
	receivedBytes		integer	Кількість отриманих байт

Продовження таблиці 2.1

	receivedPackets		integer	Кількість отриманих пакетів
	updated_at		timestamp	Дата редагування
	created_at		timestamp	Дата створення

На рисунку 2.3 зображена структурна схема бази даних.

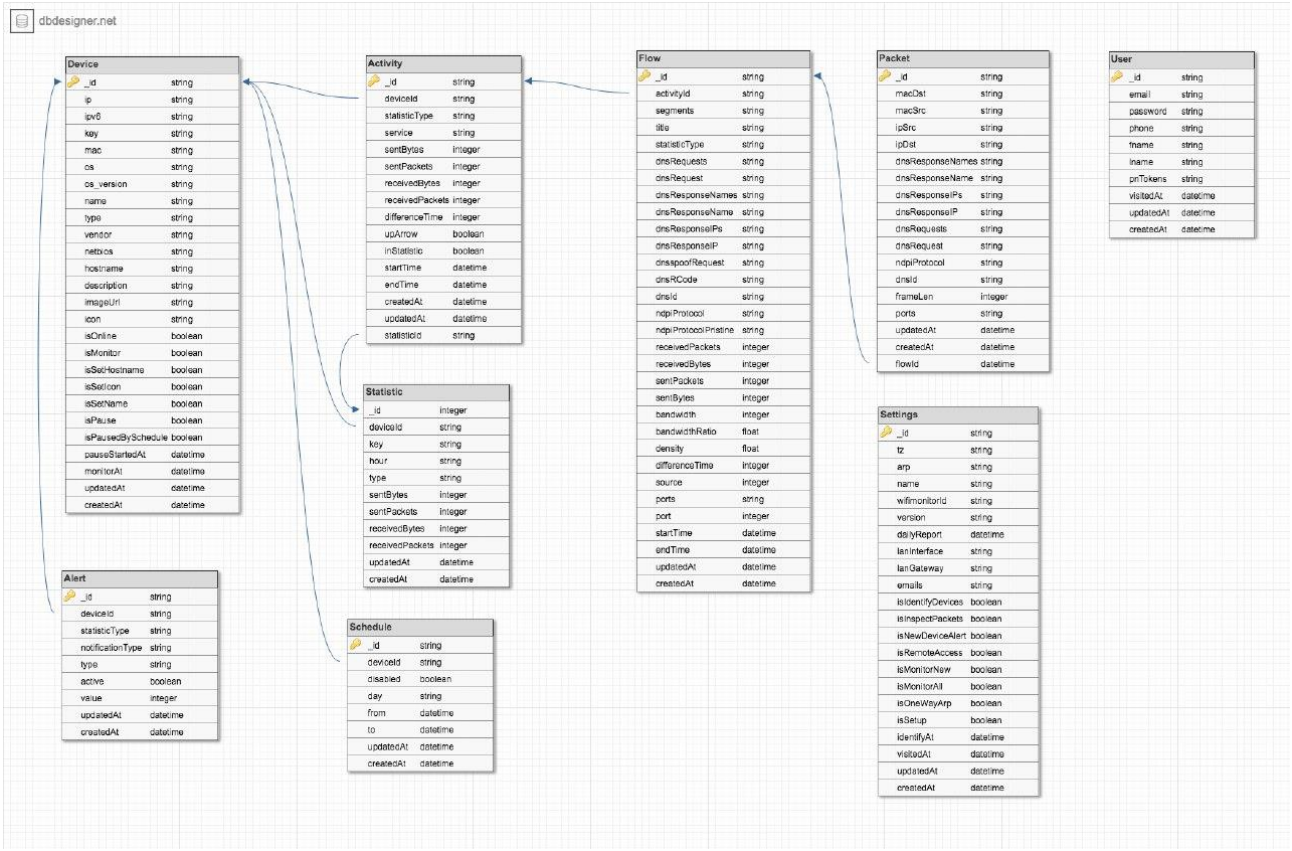


Рисунок 2.3 – Схема структурна бази даних

Висновок до розділу

В розділі описані вхідні дані програмно-апаратного комплексу, джерела вхідних даних. Описані вихідні дані, структура вихідних даних, що являють собою таблиці.

В розділі детально описані таблиці бази даних та колонки таблиць з описом даних, що зберігаються.

Наведена схема бази даних, що показує зв’язки між таблицями.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Було вирішено реалізовувати програмно-апаратний комплекс, який в свою чергу використовує Node.js, React, EcmaScript, HTML та CSS. Для розробки серверної частини була обрана мова програмування Node.js з використанням фреймворку Express. Також використовується база даних MongoDB та Redis в якості local storage.

Також були використані бібліотеки Tshark, Nmap, Arpspoof, утиліта iptables.

Iptables - це програма для користувальницьких програм, яка дозволяє системному адміністратору налаштовувати таблиці, надані брандмауером ядра Linux (реалізовані як різні модулі Netfilter), а також ланцюжки та правила, які він зберігає. В даний час для різних протоколів використовуються різні модулі та програми ядра; iptables застосовується до IPv4, ip6tables до IPv6, arptables до ARP та ebtables для Ethernet-фреймів.

Iptables вимагає підвищеної привілеї для роботи і повинна виконуватися root-користувачем, в іншому випадку він не працює. У більшості систем Linux iptables встановлюється як / usr / sbin / iptables і документується на його сторінках, які можна відкрити за допомогою iptables, коли встановлено. Він також може бути знайдений в / sbin / iptables, але оскільки iptables більше схожий на службу.

Термін iptables також зазвичай використовується для включення до компонентів рівня ядра. x_tables - це назва модуля ядра, що несе спільну частину коду, що використовується всіма чотирма модулями, що також забезпечує API, що використовується для розширень; Згодом Xtables більш-менш використовується для позначення всієї архітектури брандмауера (v4, v6, arp і eb).

TShark - аналізатор мережевого протоколу. Він дозволяє записувати дані пакету з активної мережі або читати пакети з раніше збереженого файлу збору, або друкувати декодовану форму цих пакетів до стандартного виводу або записувати пакети до файлу. Нативний формат файлу захоплення TShark - це

формат pcap, який також є форматом, який використовує tcpdump та інші інші інструменти.

Без будь-яких параметрів, TShark буде працювати так само, як tcpdump. Він використовуватиме бібліотеку pcap для захоплення трафіку з першого доступного мережевого інтерфейсу та відображає стислий рядок на стандартному виводі для кожного отриманого пакета.

Якщо запустити з параметром -r, вказавши файл знімка, з якого слід читати, TShark знову буде працювати так само, як tcpdump, читаючи пакети з файлу та показуючи стислий рядок у стандартному виводі для кожного прочитаного пакета. TShark здатний виявляти, читати та записувати ті ж файли захоплення, які підтримуються Wireshark. Вхідний файл не потребує певного розширення файлу; формат файлу та додатковий стиснення gzip будуть автоматично виявлені.

Підтримка стиснення файлів використовує (і тому вимагає) бібліотеку zlib. Якщо бібліотека zlib не присутня під час складання TShark, її можна буде скомпілювати, але в результаті програма не зможе прочитати стиснуті файли.

При відображенні пакетів на стандартному виводі, за замовчуванням, TShark записує стислий рядок, що містить поля, зазначені в файлі налаштувань (які також є полями, відображеними у панелі списку пакетів у Wireshark), хоча, якщо він записує пакети, коли він захоплює їх , а не писати пакети з збереженого файлу збору, він не покаже поле "номер кадру". Якщо вказано параметр -V, він замість цього записує перегляд деталей пакета, відображаючи всі поля всіх протоколів у пакеті. Якщо вказано параметр -O, він відображатиме лише повну інформацію про вказані протоколи та покаже лише верхню строку деталізації для всіх інших протоколів. Використовуйте висновок "tshark -G-протоколів", щоб знайти абревіатуру зазначених протоколів. Якщо опція -P вказана за допомогою параметрів -V або -O, буде показано як підсумковий рядок для всього пакета, так і деталі.

Пакет захоплення виконується за допомогою бібліотеки pcap. Ця бібліотека підтримує вказівку виразу фільтра; Пакети, які не відповідають цьому фільтру,

відкидаються. Параметр -f використовується для визначення фільтра захоплення. Синтаксис фільтра захоплення визначається бібліотекою pcap; цей синтаксис відрізняється від синтаксису фільтру читання, описаного нижче, і механізм фільтрації обмежений своїми можливостями.

Nmap (Network Mapper) - безкоштовний сканер безпеки з відкритим вихідним кодом, спочатку написаний Гордоном Ліоном (також відомий його псевдонімом Федором Васковічем), який використовувався для виявлення хостів та сервісів в комп'ютерній мережі, створюючи "карту" мережі. Щоб досягти своєї мети, Nmap надсилає спеціально сформовані пакети до цільових хостів, а потім аналізує відповідь.

Програмне забезпечення надає ряд функцій для зондування комп'ютерних мереж, включаючи виявлення та виявлення сервісів та операційну систему. Ці можливості розширюються за допомогою сценаріїв, які забезпечують більш просунуте виявлення служб, виявлення вразливостей та інших функцій. Nmap може адаптуватися до умов мережі, включаючи затримку при скануванні. Користувацька спільнота Nmap продовжує розробляти та вдосконалювати інструмент.

Nmap був заснований як утиліта, що підтримує Linux, але портована для Windows, Solaris, HP-UX, BSD (включаючи macOS), AmigaOS та IRIX. Linux є найпопулярнішою платформою, яка слідкує за Windows.

ARP-spoofing - це техніка, за допомогою якої зловмисник надсилає (підроблені) повідомлення про протокол розпізнавання адрес (ARP) на локальну мережу. Як правило, мета полягає в тому, щоб зв'язати MAC-адресу зловмисника з IP-адресою іншого хоста, наприклад, шлюз за замовчуванням, тому що будь-який трафік, призначений для цієї IP-адреси, повинен бути надісланий атакуючому.

Підроблення ARP може дозволити зловмиснику перехопити фрейми даних у мережі, змінювати трафік або зупиняти весь трафік. Часто атака використовується як відкриття для інших атак, таких як відмова в обслуговуванні, людина в середині або атаки викрадення сеансу.

Атака може використовуватися тільки в мережах, що використовують ARP, і обмежується тим, що зловмисник повинен отримати прямий доступ до сегменту локальної мережі для атаки.

React – є бібліотекою JavaScript для створення інтерфейсів коистувача. Вона підтримується Facebook і спільнотою окремих розробників та компаній.

React можна використовувати як базу при розробці single-page або мобільних додатків. Програми Complex React зазвичай вимагають використання додаткових бібліотек для управління державою, маршрутизації та взаємодії з API.

React був створений Йорданом Уолке, інженером із програмного забезпечення на Facebook. На нього вплинув XHP, компонент HTML для PHP. Він був вперше розгорнуто на новинах Facebook в 2011 році, а пізніше на Instagram.com в 2012 році. Він був відкритим на OJonf США в травні 2013 року.

React Native, який дає змогу реалізовувати реальні розробки Android, iOS та UWP за допомогою React, був оголошений у Facebook React.js Conf у лютому 2015 року та відкритий у березні 2015 року.

18 квітня 2017 року Facebook оголосив React Fiber, новий алгоритм базової бібліотеки React для створення користувацьких інтерфейсів. React Fiber повинен був стати основою для будь-яких подальших вдосконалень та розробки функцій Framework React.

Ще однією помітною особливістю є використання "віртуальної моделі об'єктів документів" або "віртуального DOM". React створює кеш-пам'ять даних у пам'яті, обчислює отримані відмінності, а потім оновлює відображений DOM веб-переглядач ефективно. Це дозволяє програмісту писати код так, ніби вся сторінка відображається на кожну зміну, а бібліотеки React відтворюють тільки ті компоненти, які дійсно змінюються.

ECMAScript – це специфікація мов сценаріїв, стандартизована ECMAScript International. Вона використовується програмами для включення сценаріїв на стороні клієнта. На специфікації впливають такі мови програмування, як Self, Perl, Python, Java і т.д. Мови, такі як JavaScript, Jscript та ActionScript, регулюються цією специфікацією.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ньому відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам, — функції як об'єкти першого рівня, об'єкти як списки, каррінг (currying), анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має С-подібний синтаксис, але в порівнянні з мовою С має такі корінні відмінності – об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів, функції як об'єкти першого класу, обробка винятків, автоматичне приведення типів, автоматичне прибирання сміття, анонімні функції. Семантика мови схожа з семантикою мови Self [4].

HTML - стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

Багато років тому, HTML був абревіатурою, яка використовувалася виключно старшими розробниками програмного забезпечення та технічними освіченими студентами. Коли світ стає більш оцифрованим, значно підвищився попит на навички кодування HTML. Для деяких HTML може звучати як іноземний термін. Це означає гіпертекстову мову розмітки. Мова гіпертекстової розмітки - це мова, яка використовується на веб-сайтах для відображення шрифтів, кольорів, посилань та зображень відвідувачів сайту. У межах мови існує кілька тегів, які використовуються для того, щоб дозволити творцю сайту налаштовувати спосіб перегляду сайту. Ці теги та коди потім читаються веб-переглядачами, які відображають веб-сайт відповідно до специфікацій, перерахованих у кодах HTML. Відвідувачі сайту, а потім переглядають окремі сторінки в межах сайту, створені розробником, не бачачи переліку кодів і тегів.

Транспортування в мережі.

HTML документи можуть бути транспортовані так само як і будь-які інші файли (наприклад, за допомогою протоколів FTP, TCP), проте зазвичай вони

транспортуються із сервера за допомогою протоколу HTTP або по електронній пошті [5].

HTTP.

HTTP функціонує як протокол запиту-відповіді у моделі обчислень клієнт-сервер. Наприклад, веб-браузер може бути клієнтом, а програма, запущена на комп'ютері, на якому розміщено веб-сайт, може бути сервером. Клієнт подає на сервер запит HTTP-запиту. Сервер, який надає ресурси, такі як файли HTML та інший вміст, або виконує інші функції від імені клієнта, повертає відповідне повідомлення клієнту. Відповідь містить інформацію про статус завершення про запит, а також може містити запитуваний вміст у своєму повідомленні.

Веб-браузер є прикладом користувацького агента (UA). Інші типи користувальницького агента включають програмне забезпечення для індексування, яке використовуються пошуковими провайдерами (веб-сканерами), голосовими браузерами, мобільними програмами та іншим програмним забезпеченням, яке має доступ, споживає або відображає веб-вміст.

HTTP призначений для того, щоб дозволити проміжним елементам мережі покращувати або активувати зв'язок між клієнтами та серверами. Веб-сайти з високим трафіком часто користуються серверами веб-кешу, які постачають вміст від імені серверів на вищому рівні, щоб поліпшити час відгуку. Веб-браузери кешують раніше доступні веб-ресурси та повторно використовують їх, коли це можливо, щоб зменшити мережевий трафік. HTTP-проксі-сервери в межах приватної мережі можуть полегшити спілкування для клієнтів без глобальної маршрутизації, шляхом передачі повідомлень із зовнішніми серверами.

HTTP - протокол прикладного рівня, розроблений в рамках пакету Інтернет-протоколів. Його визначення передбачає основний і надійний протокол транспортного рівня, та протокол управління передачею (TCP). Проте HTTP може бути адаптований для використання ненадійних протоколів, таких як Протокол обробки даних (UDP) користувачів, наприклад, у протоколі HTTPU та Простий протокол розпізнавання служб (SSDP).

Ресурси HTTP ідентифікуються та розміщуються в мережі уніфікованими локаторами ресурсів (URL-адреси), використовуючи схеми уніфікованого

ідентифікатора ресурсів (URI) http і https. URI та гіперпосилання у документах HTML утворюють взаємопов'язані гіпертекстові документи.

HTTP / 1.1 - це перегляд вихідного HTTP (HTTP / 1.0). У HTTP / 1.0 для кожного запиту ресурсу створюється окреме з'єднання з тим самим сервером. HTTP / 1.1 може повторно використовувати зв'язок кілька разів, щоб завантажувати зображення, сценарії, таблиці стилів тощо після того, як сторінка була доставлена. Отже, HTTP / 1.1 зв'язку відчувають меншу затримку, оскільки встановлення TCP-з'єднань створює значні накладні витрати [6].

CSS – спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

Переваги:

Інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;

Різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;

Сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;

Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної [7].

Node.js – це відкрите, крос-платформенне середовище виконання середовища JavaScript, яке виконує код JavaScript за межами веб-переглядача. Як правило, JavaScript використовується насамперед для сценаріїв на стороні клієнта, в яких сценарії, написані на JavaScript, вставляються в HTML-сторінку веб-сторінки та запускаються на стороні клієнта за допомогою двигуна JavaScript у веб-браузері користувача. Node.js дозволяє розробникам використовувати JavaScript для написання інструментів командного рядка та серверного скриптів для скриптів сценаріїв на сервері для створення динамічного вмісту веб-сторінок, перш ніж сторінка надсилатиметься в веб-браузері користувача. Отже, Node.js являє собою парадигму "JavaScript скрізь", що об'єднує розробку веб-програм навколо однієї мови програмування, а не різних мов для скриптів на стороні сервера та клієнта.

Node.js дозволяє створювати веб-сервери та інструменти для роботи з мережею, використовуючи JavaScript і збірку "модулів", які обробляють різні функціональні можливості сервера. Модулі передбачаються для введення / виводу файлової системи, мережевих мереж (DNS, HTTP, TCP, TLS / SSL або UDP), двійкові дані (буфери), функції криптографії, потоки даних та інші основні функції. Модулі Node.js використовують API, призначений для зменшення складності написання серверних програм.

Хоча спочатку модульна система була заснована на шаблоні модуля commonjs, останнім часом введення модулів в специфікації ECMAScript змінювало напрямок використання модулів ECMAScript в Node.js замість цього.

Node.js забезпечує керування подіями веб-серверами, що дозволяє створювати швидкі веб-сервери в JavaScript. Розробники можуть створювати

високомасштабні сервери без використання різання, використовуючи спрощену модель подій, керованих програмами, які використовують зворотні виклики, щоб сигналізувати про завершення завдання. Node.js пов'язує простоту мови скриптів (JavaScript) з потужністю мережевого програмування Unix.

Node.js був побудований на движку Google V8 JavaScript, оскільки він був відкритим за ліцензією BSD. Він надзвичайно швидкий і володіє основою Інтернету, такими як HTTP, DNS, TCP. Крім того, JavaScript - це добре відома мова, що робить Node.js відразу доступним для всієї спільноти веб-розробників.

Node.js зареєстрований з операційною системою, щоб ОС сповістила про з'єднання та видає зворотний виклик. У середовищі виконання Node.js кожне підключення - це невеликий розмір купу. Традиційно, відносно важкоатлетичних операцій ОС або потоків обробляються кожне з'єднання. Node.js використовує цикл подій для масштабованості замість процесів або потоків. На відміну від інших серверів, керованих подіями, цикл подій Node.js не повинен називатися явним чином. Замість цього визначаються зворотні виклики, і сервер автоматично входить до циклу подій у кінці визначення зворотного виклику. Node.js виходить з циклу подій, коли немає додаткових зворотних викликів, які потрібно виконати.

MongoDB – це крос-платформа, орієнтована на документи база даних, яка забезпечує високу продуктивність, високу доступність і легку масштабованість. MongoDB працює над концепцією колекціонування та документацією.

База даних - це фізичний контейнер для колекцій. Кожна база отримує власний набір файлів у файловій системі. Один сервер MongoDB, як правило, має кілька баз даних.

Колекція - це група документів MongoDB. Це еквівалент таблиці таблиці СУБД. Колекція існує в єдиній базі даних. Колекції не впроваджують схему. Документи в колекції можуть мати різні поля. Як правило, всі документи в колекції мають схожі або пов'язані цілі.

Документ являє собою набір пар ключ-значення. Документи мають динамічну схему. Динамічна схема означає, що документи в тій же колекції не

повинні мати однаковий набір полів або структури, а загальні поля у документах колекції можуть містити різні типи даних.

3.2 Вимоги до технічного забезпечення

Загальні вимоги

Для правильної роботи даної програми до складу технічних засобів повинні входити такі компоненти.

Для веб-додатку.

а) комп'ютер з такою конфігурацією:

1) процесор – 1 Ггц, 2 ядра ЦП або краще;
2) оперативна пам'ять не менш ніж 1048 Мб;
3) не менше ніж 2 ГБ ПЗУ;
4) 10 Мбіт/сек доступу до мережі Інтернет;

б) додатково має бути встановлене таке програмне забезпечення:

1) операційна система;
2) база даних MongoDB;
3) Node.js 6.10.3 і вище;
4) веб сервер Apache.

Клієнт, на якому запускається веб-сторінка застосунку має задовольняти таким вимогам:

1) наявність браузеру;
2) операційна система Windows XP або вище;
3) доступ до інтернету.

3.3 Архітектура програмного забезпечення

Діаграма послідовності

Діаграма послідовності для процесу здійснення атаки на інформаційну систему та визначення типу атаки після сканування системи на вразливості, що представлена на рисунку 3.1 та в таблиці 3.1

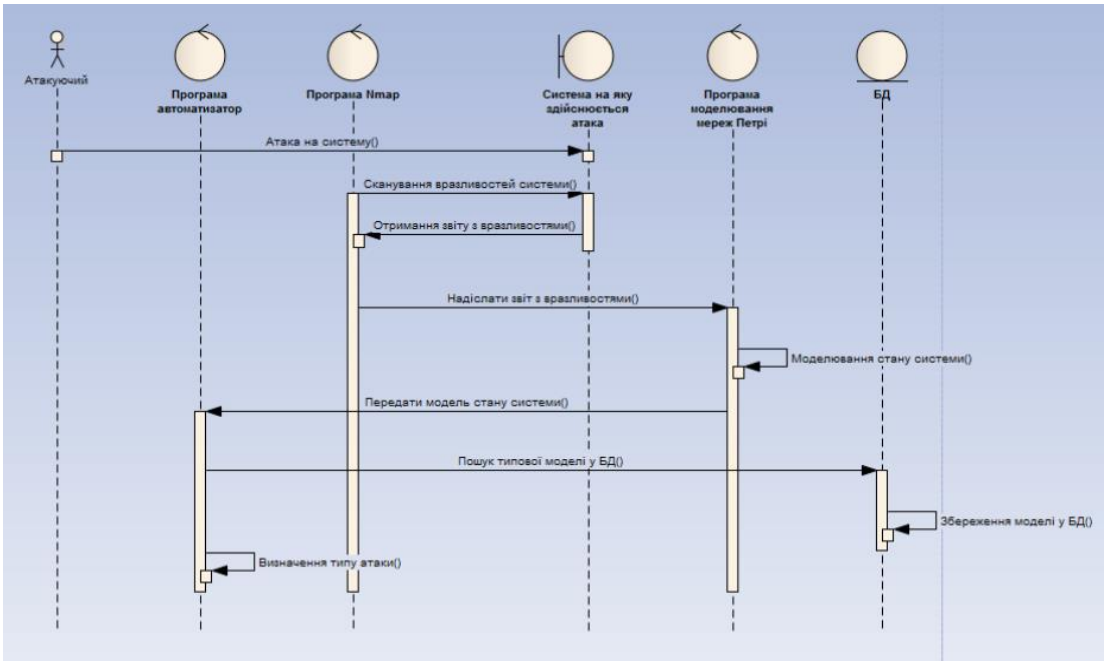


Рисунок 3.1 – Діаграма послідовності

Таблиця 3.1 – Об’єкти діаграми послідовності

Клас	***Відповідальність***
Атакуючий	***Сканування мережі та пошук потрібної системи для атаки, сканування вразливостей, вибір оптимального сценарію атаки, налаштування та підготовка експлоітів, запуск експлоітів та здійснення атаки.***
Інформаційна система на яку здійснюється атака	***Інформаційна система на яку здійснюється атака з встановленою опепаційною системою Windows XP і з налаштованими для прийняття атаки параметрами.***
Програма автоматизатор	***Зберігає у БД модель стану системи з програми моделювання мереж Петрі; Укріпляє та налоштовує ІС на яку здійснюється атака; Порівнює поточні стани системи з моделями які вже існують в БД; Визначає тип атаки***
Програма моделювання мереж Петрі	***Візуалізує та моделює стан системи після атаки за допомогою звіту з вразливостями***
Сервер БД	***Сервер БД дозволяє зберігати стани системи після атаки. Збережені моделі на сервері БД, працюючи безпосередньо з таблицями, серед інших виконують такі функції: Зберегти модель стану системи ; Визначити тип атаки за допомогою порівнянь поточного та існуючого станів.***
Програма Nmap	***Сканує систему на вразливості***

Інцидент порушення безпеки

Нехай маємо підмережу, що складаєтсья з незалежних пристроїв і комутатору (маршрутизатор у нашому випадку). Кожен пристрій довільно і в будь-який момент робить запит. Схема для конкретного пристрою виглядає наступним чином (Рисунок 3.2):

Рисунок 3.2 – Схема комунікації пристрою підмережі з роутером за нормальних умов

На момент запиту починається обмін пакетом між пристроєм і маршрутизатором. Роутер, у свою чергу, передає запит до World Wide Web і чекає відповіді. Після отримання відповіді маршрутизатор надсилає його запитуючому пристрою. Пристрій зі свого боку збирає набір пакетів, інтерпретуючи його в зручний для комп'ютера формат, а потім показує його для користувача. Процес інтерпретації починається після отримання останнього пакета відповідей. Коли почалася повномасштабна атака, це означає, що певні пакети були заздалегідь навмисно фальсифіковані або замінені в процесі передачі. У ідеальній ситуації брандмауер операційної системи буде розпізнавати атаку і заблокує його. Якщо брандмауер не знайшов загрози, яка пройшла з відповіддю на сервер, то пристрій заражений і відповідальність за подальше визначення атаки падає на антивірусне рішення. Антивірус при аналізі стану системи визначить або не визначить цю атаку.

Процес реагування на інцидент і вирішення проблеми кібербезпеки

Прогнозоване рішення - програмно-апаратний комплекс. Термін «апаратний» використовується тому, що алгоритм реалізований на окремому пристрої, який підключений до підмережі для декількох пристроїв. Користувач

може одночасно активувати захист для кожного окремо або для всіх пристроїв. Під час активації захисту відбувається атака "man-in-the-middle" на рівні маршрутизації трафіку. Тоді весь трафік проходить через підключений пристрій, що означає, що в цьому випадку схема взаємодії для певного пристрою буде наступною (рисунок 3.3):

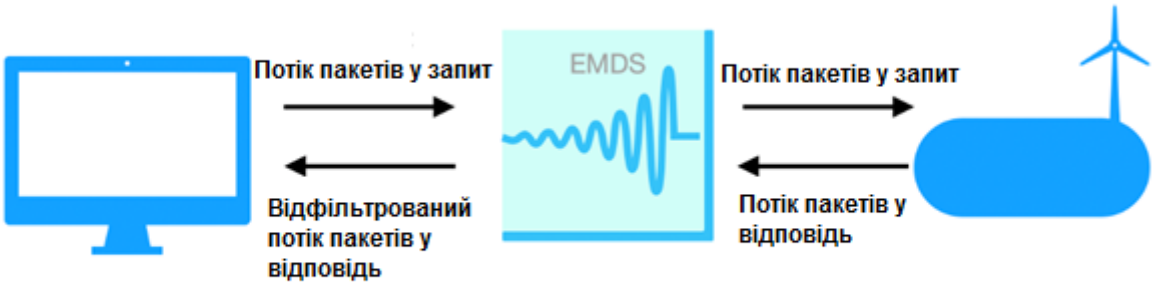


Рисунок 3.3 – Схема комунікації пристрою підмережі з роутером після впровадження програмно-апаратного комплексу

Тепер кожен пристрій бачить програмно-апаратний комплекс як маршрутизатор, а самий маршрутизатор сприймає програмно-апаратний комплекс як остаточний пристрій. У цьому випадку програмно-апаратний комплекс переслідує політику взяття на себе відповідальності за постачання пакетів з виходом з пристроїв на маршрутизатор та пакети відповідей від маршрутизатора до пристрою. Отже, тепер, коли трафік проходить через програмно-апаратний комплекс, існує можливість аналізувати кожен з пакетів. Для комплексного аналізу зберігання пакетів та їх вмісту в базі даних є обов'язковим.

Аналіз пакетів складається з 2-фазної перевірки. На першому етапі була проведена перевірка на сигнатури вже відомих атаках. Незважаючи на швидкий розвиток технології, основна кількість атак виявляється за допомогою методу сигнатур. Точність роботи всіх методів буде залежати від якості сигнатури.

На цьому етапі програмно-апаратний комплекс виявить вторгнення в реальному часі. Це означає, що він буде працювати, перш за все, як система виявлення вторгнень (IDS). По-друге, він забезпечить захист від вторгнень - датчик запобігання вторгненню (IPS), а також буде контролювати безпеку

мережі - Network Security Monitor (NSM). Система відслідковує критичні характеристики мережі в режимі реального часу та генерує сигнал тривоги, коли виявляє дивну подію, яка може вказувати на загрозу. Приклади таких показників включають обсяг трафіку, використання пропускної здатності та використання протоколу. Перевірка мережевого трафіку включає потужні та широкі правила та мову сигнатур. Цей метод перевірки можливий на стадії проходження через програмно-апаратний комплекс, і в разі загрози, шкідливу відповідь не буде пропущено до кінцевого пристрою.

На другому етапі проводиться поведінковий аналіз. Цей метод заснований на послідовності різних запитів. Аналіз поведінки в мережі - це можливість ідентифікувати трафік, який є звичайним для щоденного мережевого трафіку. Іншими словами, це спроба виявити перешкоди в мережі, якщо трафік перевищив встановлений раніше пороговий показник. Одним із найбільш відомих порушення мережевої безпеки є напад, відомий як "розподілена відмова в обслуговуванні" (DDoS). Це серйозна загроза безпеці інтернет-провайдерів та великих мережевих інфраструктур. Давайте розглянемо цю фазу за допомогою прикладу. Нехай буде послідовність запитів, наданих у таблиці 3.2.

Таблиця 3.2 – Функції програмного забезпечення

№	Запит
1	/api/products?userId=1
2	/api/products?userId=3-2
3	/api/products?userId=-1
4	/api/products?userId=1'
5	/api/products?userId=‘1
6	/api/products?userId=1 and sleep(5)

У цьому випадку існує спроба ідентифікації SQL-ін'єкції шляхом маніпулювання характеристиками, додавання лапки та виклику функції "сну". Самі по собі ці розсіяні симптоми не містять очевидного вектора атаки, але багато хто з них, очевидно, припускають, що злочинці намагаються "розвідати" веб-додаток.

Специфікація функцій

Функції програмного забезпечення наведені в таблиці 3.3.

Таблиця 3.3 – Функції програмного забезпечення

Назва	**Примітка**
Функція: User – функція, що відповідає за користувача.	
add ()	Створення нового користувача. Викликається при реєстрації.
destroy (id: int)	Видалення користувача.
bulkDestroy (ids[]: int)	Массове видалення користувачів.
update (id: int)	Редагування користувача, спрацьовує при натисненні кнопки «Edit» на списковому представленні розділу Users.
list()	Отримання списку всіх користувачів.
get (id:int)	Отримання даних користувача.
login(email:string, password: string)	Виконання входу в систему
authenticate(auth_token)	Аутентифікація користувача за токеном. Викликається перед виконаннам кожної дії для перевірки авторизації користувача
generatePassword()	Генерація пароля. Викликається при відновленні паролю.
restorePassword(email)	Відновлення паролю.
sendEmail(user: User, subject: string, text: text)	Надсилання електронного листа користувачу.
logout(auth_token:string)	Виконання виходу із системи.

Продовження таблиці 3.3

Функція: Port – функція, що відповідає за порти системи.	
open (id: int)	Відкриття порту з відповідним номером.
close (id: int)	Закриття порту з відповідним номером.
Функція: Firewall – функція, що відповідає за порти системи.	
turnOn ()	Ввімкнення фаєрволу.
turnOff ()	Вимкнення фаєрволу.
Функція: AtackType – функція, що відповідає за сканування QR-коду	
add ()	Створення нового типу атаки.
destroy (id: int)	Видалення типу атаки.
get (id:int)	Отримання даних типу атаки.
update (id: int)	Редагування типу атаки, спрацьовує при натисненні кнопки «Edit» на списковому представленні розділу Atack types.
list()	Отримання списку всіх типів атак.
Функція: PetriNet – функція, що відповідає за відповідність певного типу атаки json-об’єкту мережі Петрі.	
add ()	Створення нової відповідності типу атаки до json-об’єкту мережі Петрі.
destroy (id: int)	Видалення типу відповідності типу атаки до json-об’єкту мережі Петрі.
get (id:int)	Отримання відповідності типу атаки до json-об’єкту мережі Петрі.
update (id: int)	Редагування відповідності типу атаки до json-об’єкту мережі Петрі, спрацьовує при натисненні кнопки «Edit» на списковому представленні розділу Atack types.
verify (file)	Перевірка відповідності стану системи існуючим в базі мережам Петрі.

3.4 Інструкція користувача

Результатом роботи є розроблений програмно апаратний комлекс аналізу вразливостей пристроїв однієї підмережі з доступом до мережі Інтернет.

Для того щоб мати можливість контролювати роботоздатність та оновлення пристроїв була розроблена панель адміністрування.

Список всіх пристроїв представлено на рисунку 3.4.

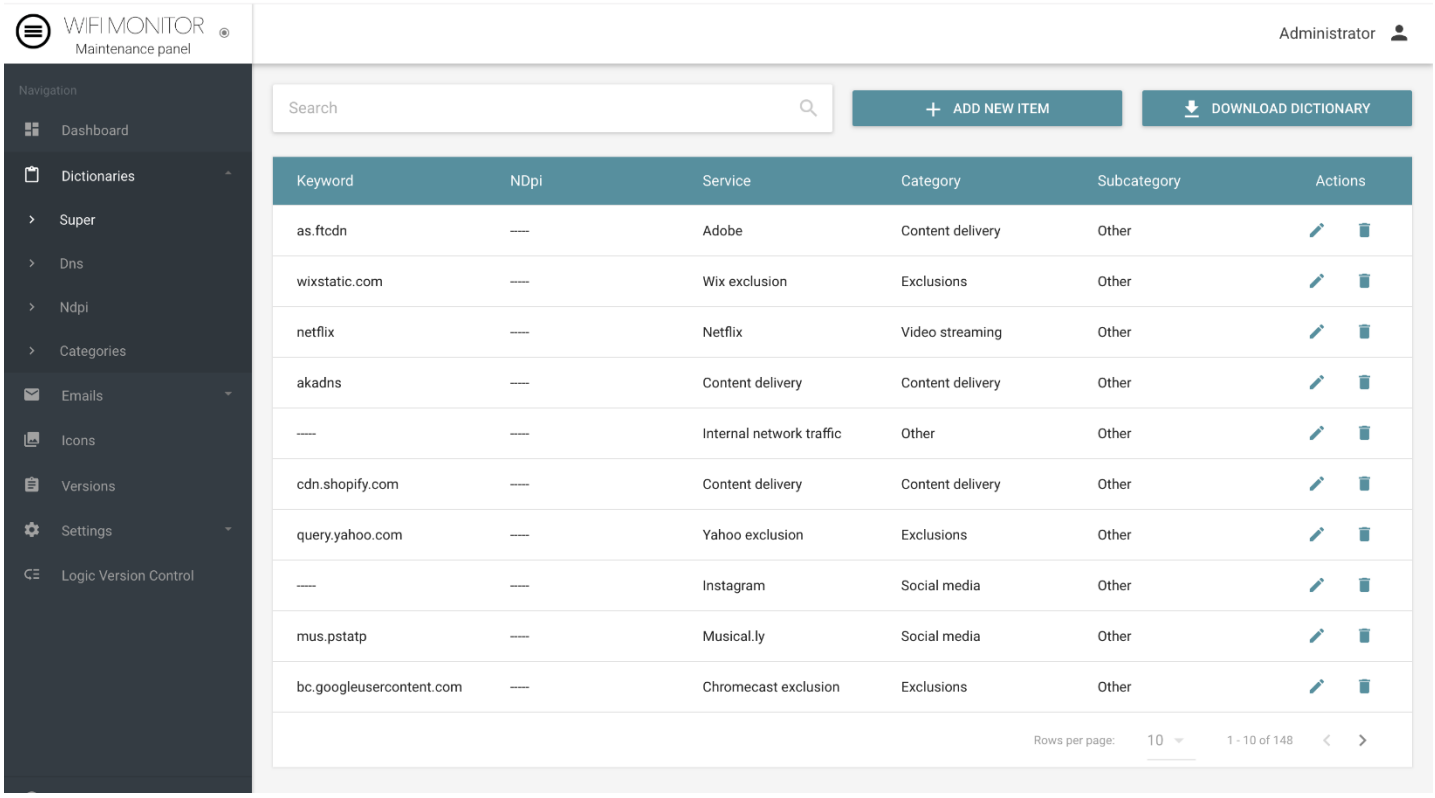


Рисунок 3.4 – Панель адміністрування. Список всіх пристроїв

Для кожного із пристроїв є можливість встановити одну з наявніх версій прошивок, оновити правила, за якими визначаються активності пристроїв підмережі, послати адміністраторові електронний звіт на пошту та доступитися до інтерфейсу окремого пристрою.

Наступним пунктом панелі адміністрування є правила, за якими визначаються активності.

Було виділено наступні категорії активностей представлені на рисунку 3.5.

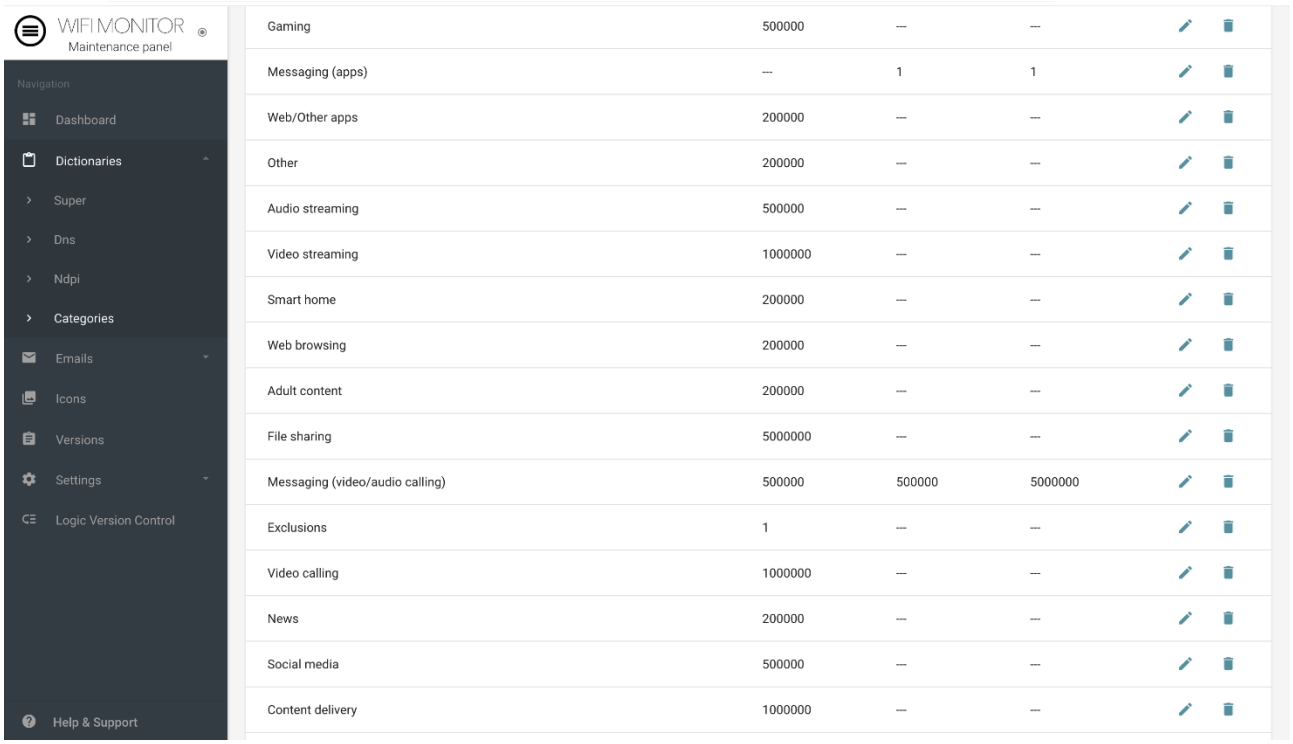


Рисунок 3.5 – Панель адміністрування. Перелік категорій активностей

Кожну категорію можна конфігурувати наступними параметрами:

- нижня границя кількості відправленних байт;
- нижня границя кількості отриманих байт;
- процент співвідношення кількості отриманих і відправлених байт для розуміння, якого типу є активність - вхідною чи вихідною;
- граничні дані для форматування активності такого типу у звіті.

Описані дані представлено на рисунку 3.6.

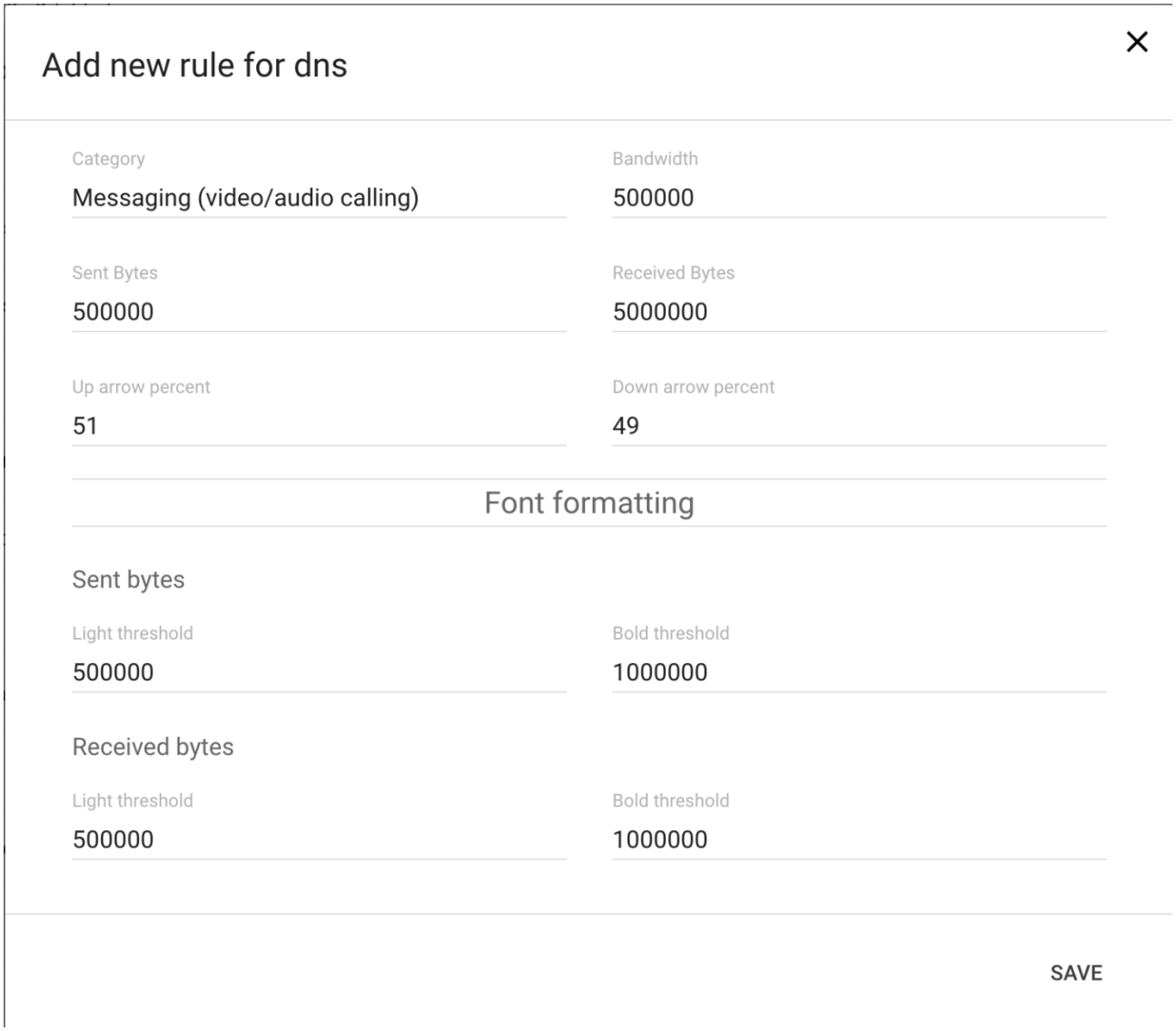


Рисунок 3.6 – Панель адміністрування. Параметри конфігурації категорії

Є 3 типи словників, що містять правила - super, dns та ndpi. У порядку такого пріоритету в процессі визначення активностей виконується правил, яке буде застосоване до того чи іншого пакету.

На рисунку 3.7 представлені деякі правила з довідника super.

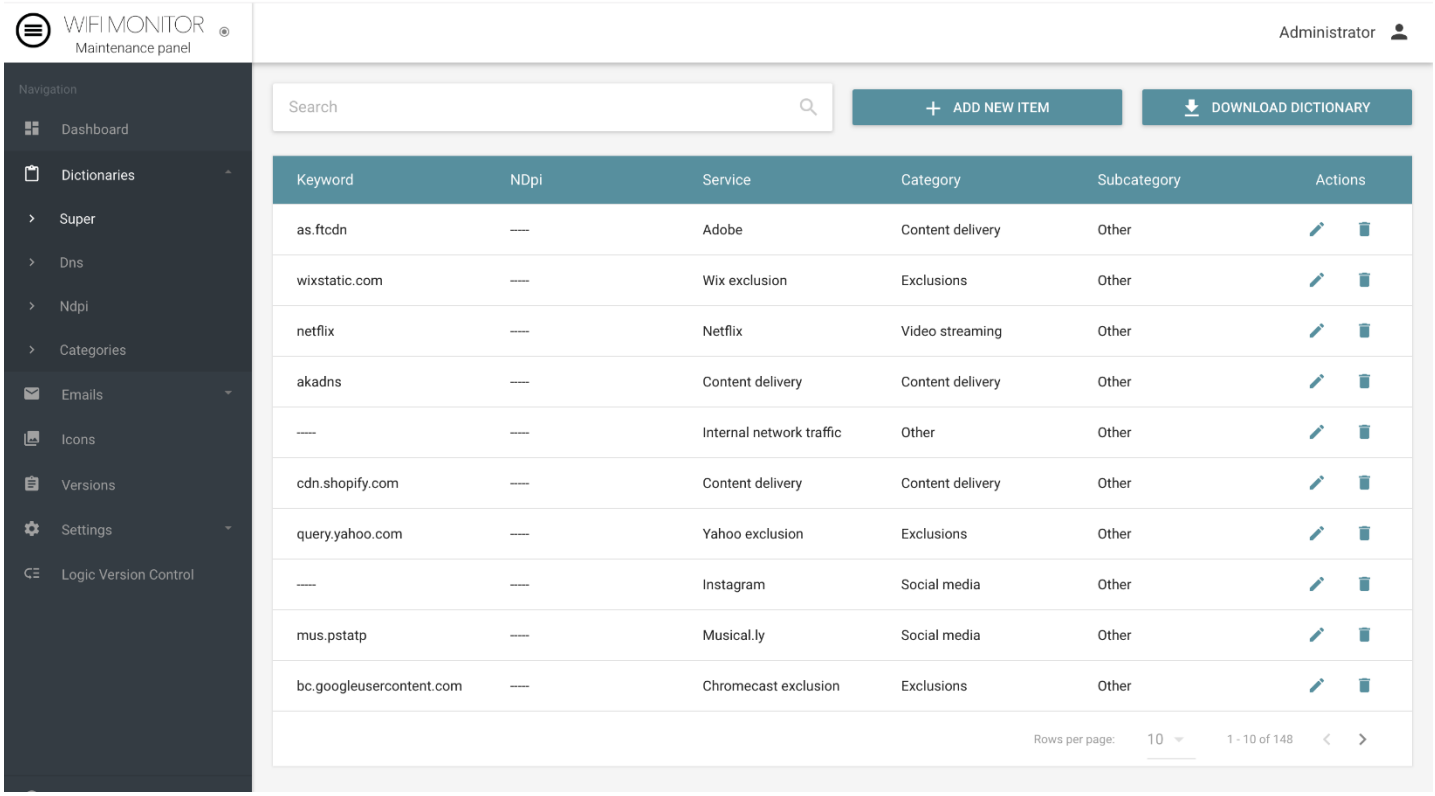


Рисунок 3.7 – Панель адміністрування. Super довідник

Правила з довідників dns та ndpi мають схожий вигляд.

Для кожного з довідників є можливість додавати нові правила. Як видно на рисунку 3.8, при створенні нового правила вказується яким чином активність буде відображена у звіті - тобто назва активності, за яким ключовим словом DNS запиту чи NDPI протоколу буде застосоване правило, а також є можливість параметризувати правило (додати обмеження) за допомогою набору значень, що містяться у пакеті. Набір параметрів видно на рисунку 3.9.

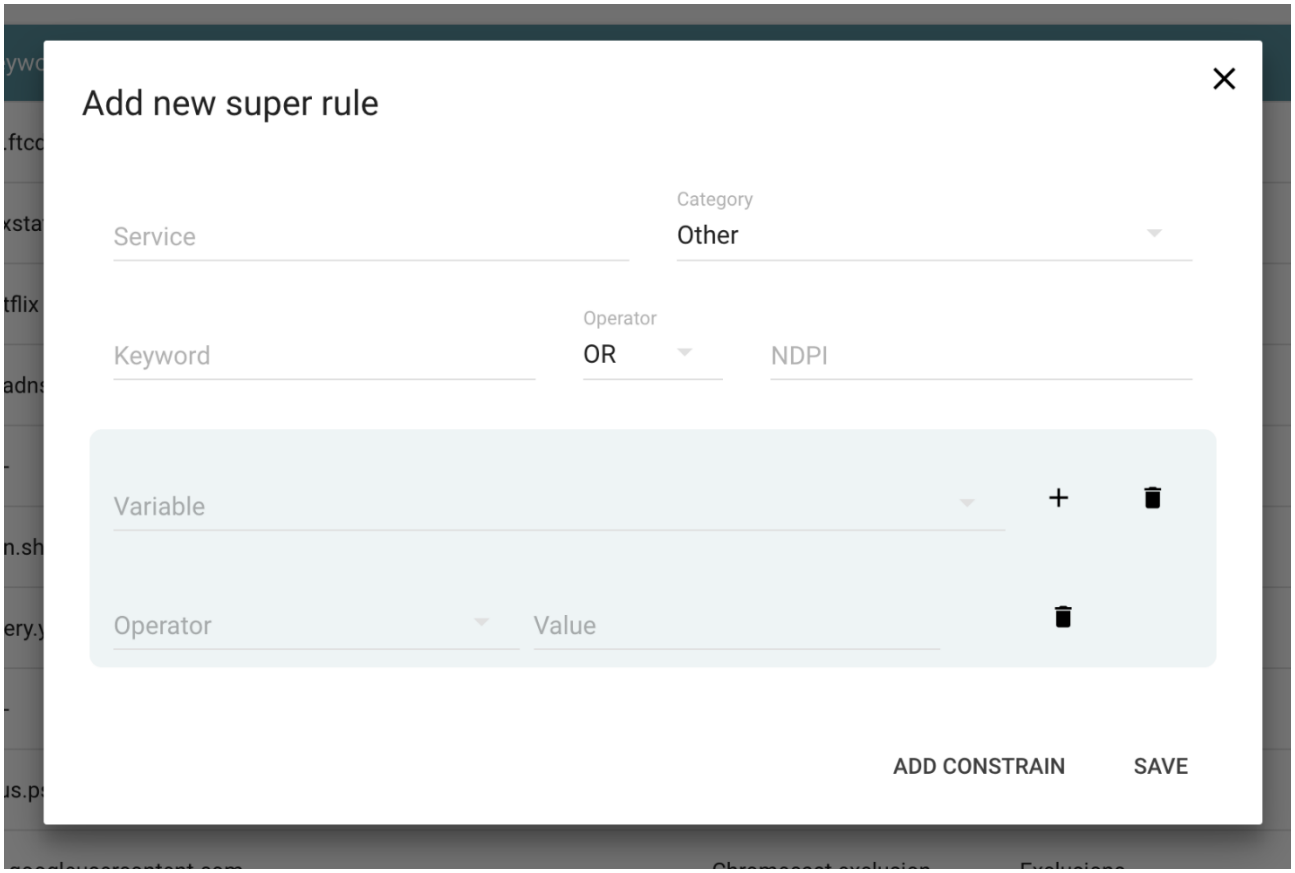


Рисунок 3.8 – Панель адміністрування. Параметризація правила

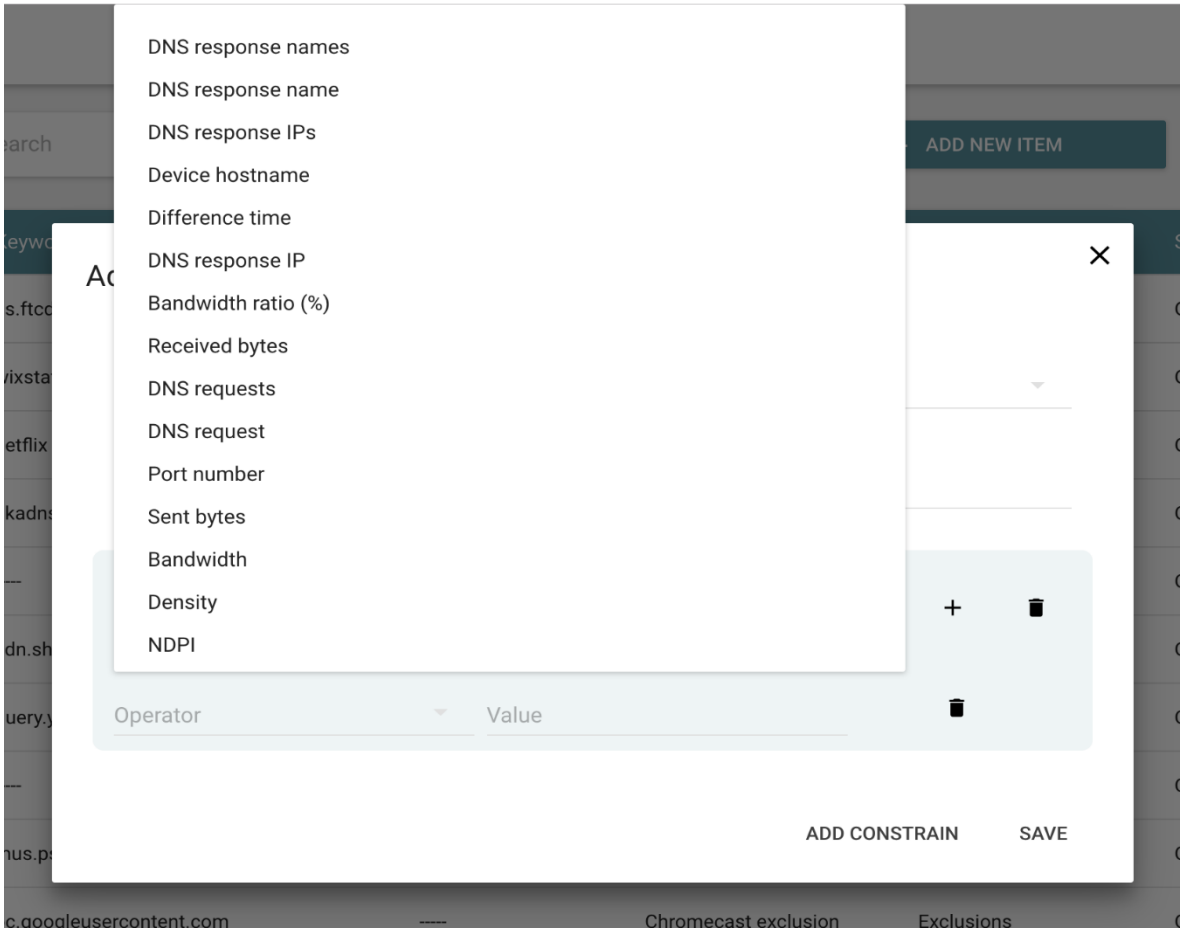


Рисунок 3.9 – Панель адміністрування. Набір параметрів пакетів

Також є можливість експортувати кожен із довідникив в csv форматі.

У розділі Email є два підпункти меню - Blocked та Unsubscribed - це перелік листів, що відправлені певним програмно-апаратним комлексом і за деяких причин не змогли бути доставленими до адресата та перелік email адрес, що відписалися від отримання листів, відповідно (рисунок 3.10).

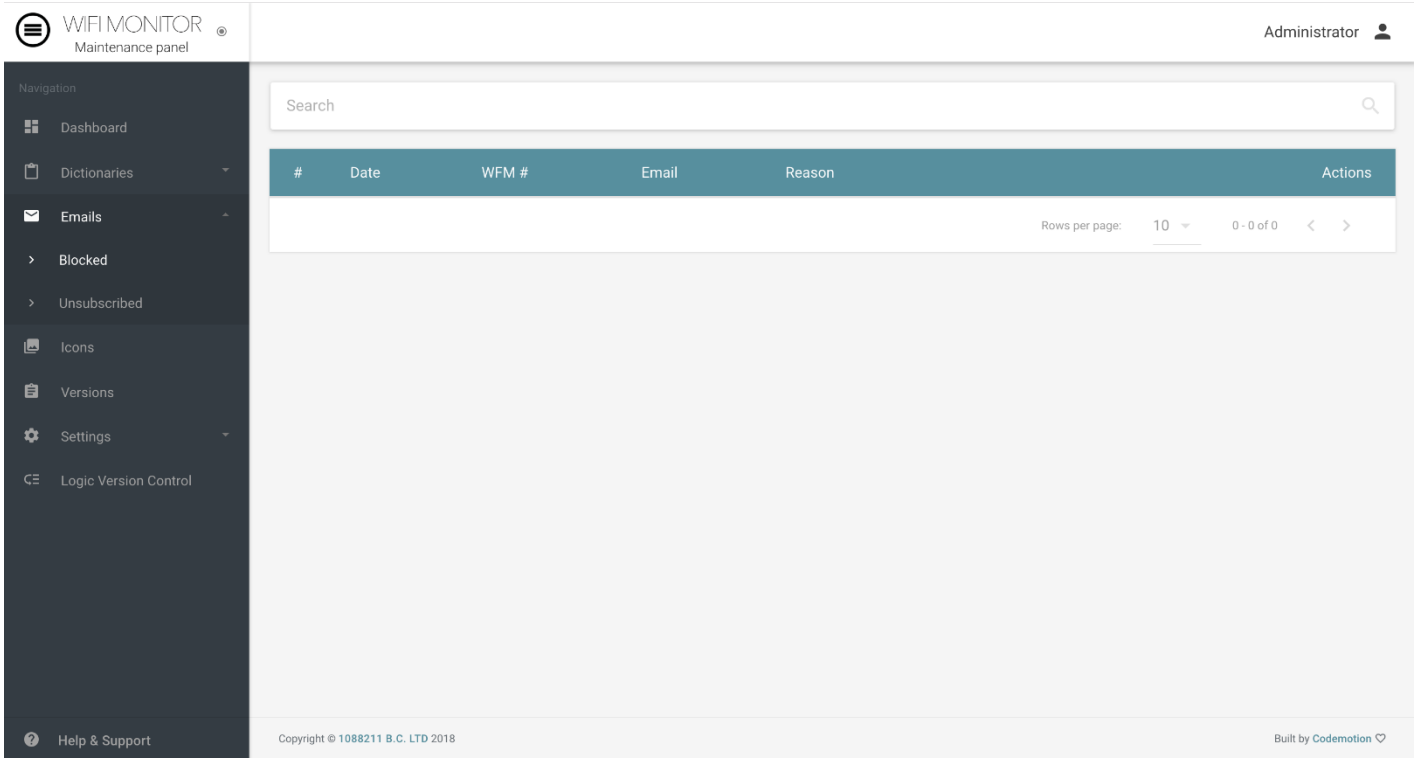


Рисунок 3.10 – Панель адміністрування. Розділ емейлів

У розділі Icons можна переглянути та відредагувати довідник, що містить інформацію по іконкам, які будуть асоційовані з пристроями підмережі для визначення їх типу для кінечного користувача. Дана реалізація представлена на рисунку 3.11.

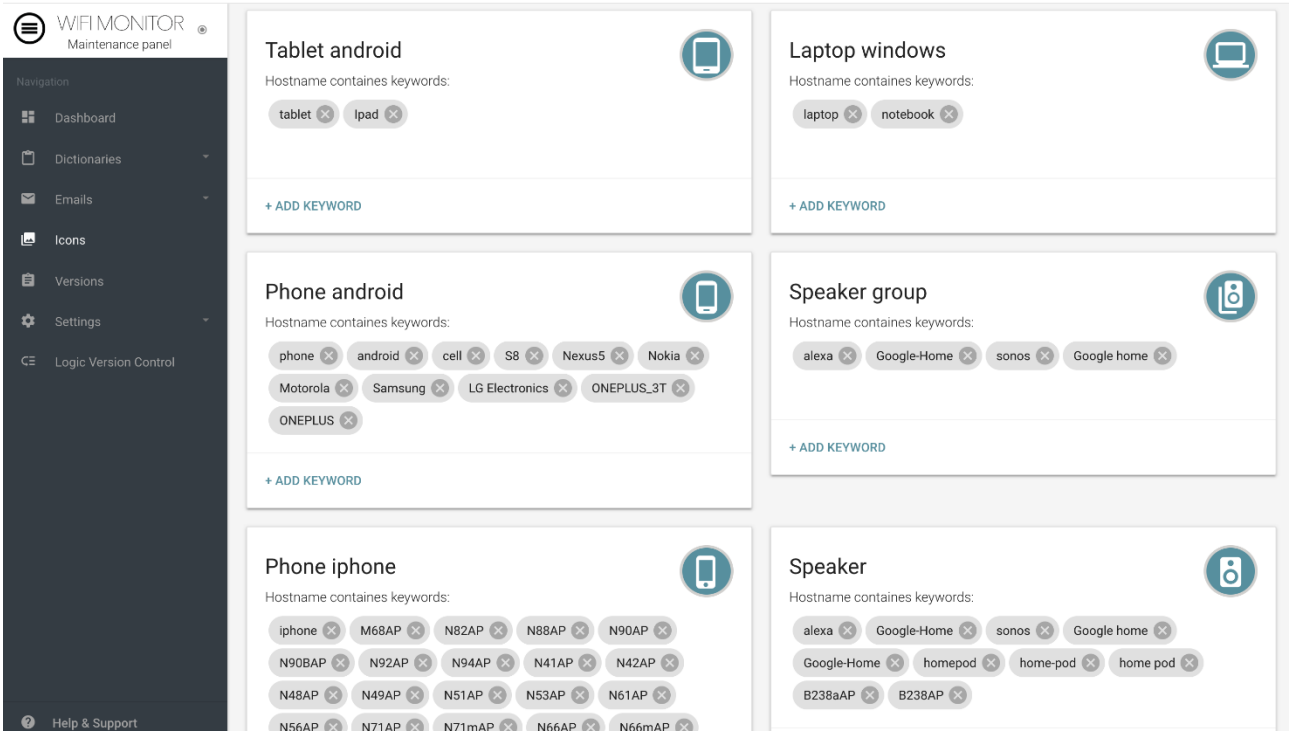


Рисунок 3.11 – Панель адміністрування. Розділ управління іконками

На рисунку 3.12 представлено розділ Versions, де надано опис змін, що входять до кожної з версій, що дозволяє чітко розуміти, послідовність покращення реалізації та появу нових функціональних можливостей.

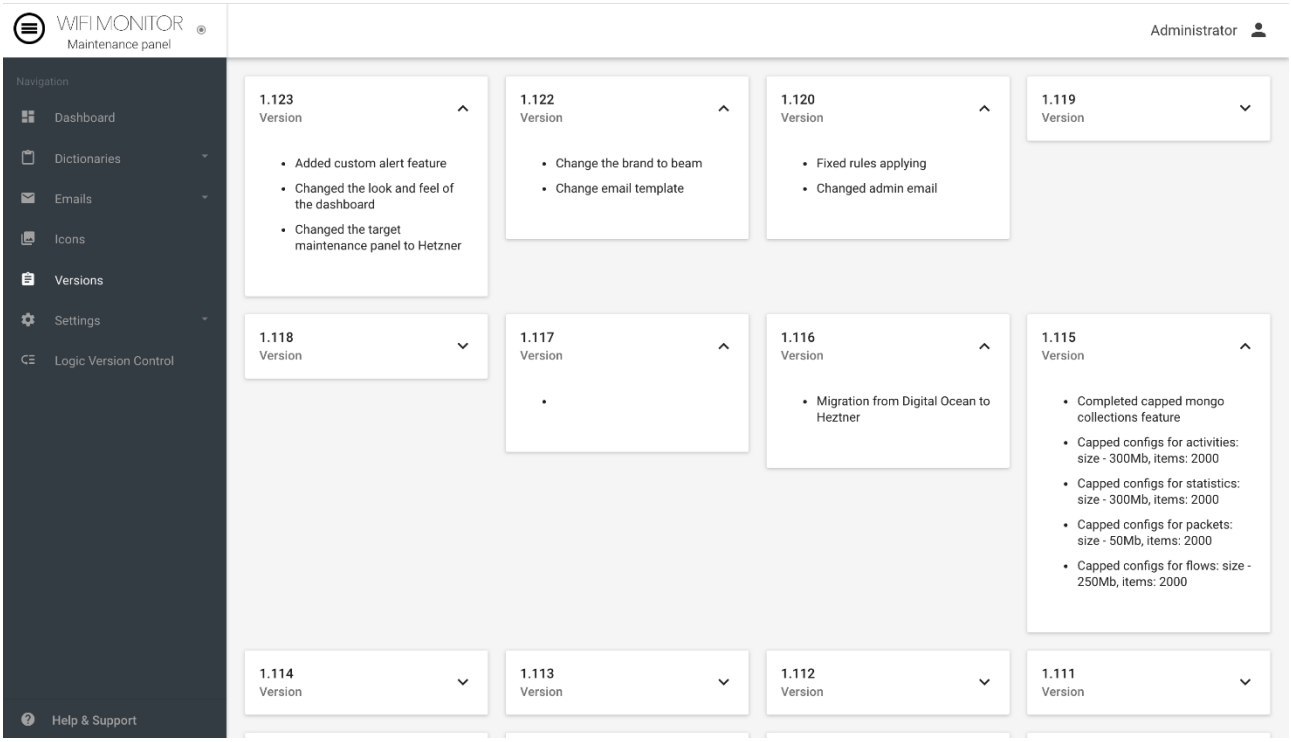


Рисунок 3.12 – Панель адміністрування. Розділ версіонування прошивок

Також є розділ контролю логіки, що дозволяє робити снепшоти стану правил, зберігати їх як окрему версію і потім повертатися до якогось стану. Дана реалізація представлена на рисунку 3.13.

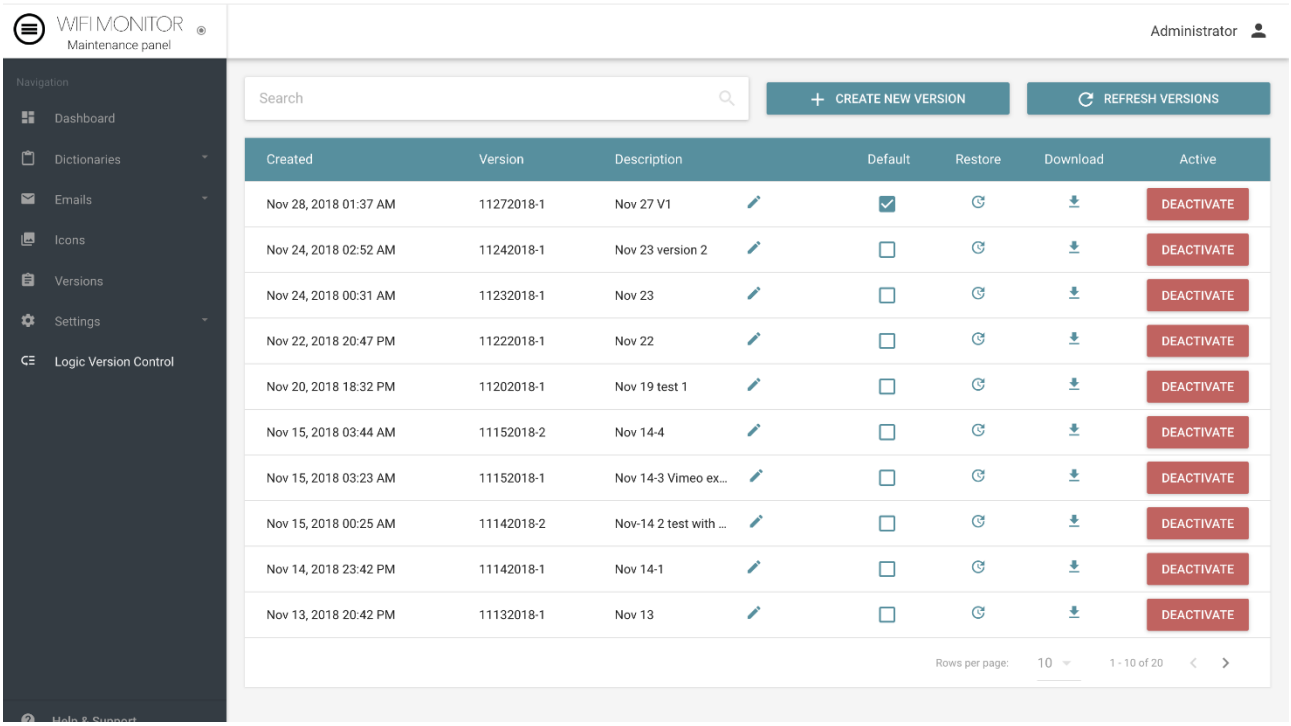


Рисунок 3.13 – Панель адміністрування. Розділ версіонування правил

Програмно-апаратний комплекс дуже простий у використанні. Достатньо вього навсього підключити його до мережі та з’єднати з роутером через мережевий інтерфейс. При підключенні його до мережі відбудеться запуск образу операційної системи. Після її запуску відбудеться ініціалізація всіх модулив операційної системи та їх запуск - Nmap, Arp-spoof, T-shark, iptables - після чого запуститься сервер з розробленим програмним забезпеченням. В свою чергу сервер відповідальний за перевірку коректності запуску всіх потрібних модулів ОС. Якщо будь-який з них не працює на фазі запуску серверу, то сервер не буде запущено. Користувач при спробі зайти на інтерфейс побачить повідомлення про помилку та інструкції щодо подальших дій для відновленяя дієспроможності програмно-апаратного комплексу.

У випадку, якщо запуск пройшов успішно користувач може зайти на інтерфейс та побачити статистику по всім пристроям за певну дату (рисунок 3.14).

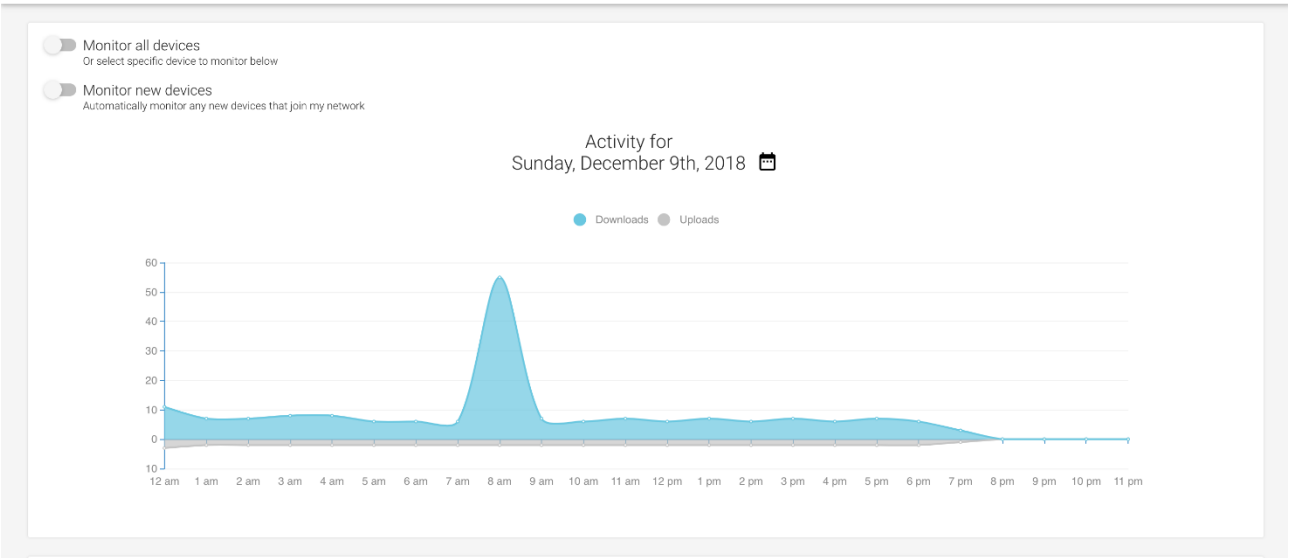


Рисунок 3.14 – Програмно-апаратний комплекс. Статистика по всім пристроям

Під статистчними даними по всім пристроям буде слідувати список всіх пристроїв, як представлено на рисунку 3.15.

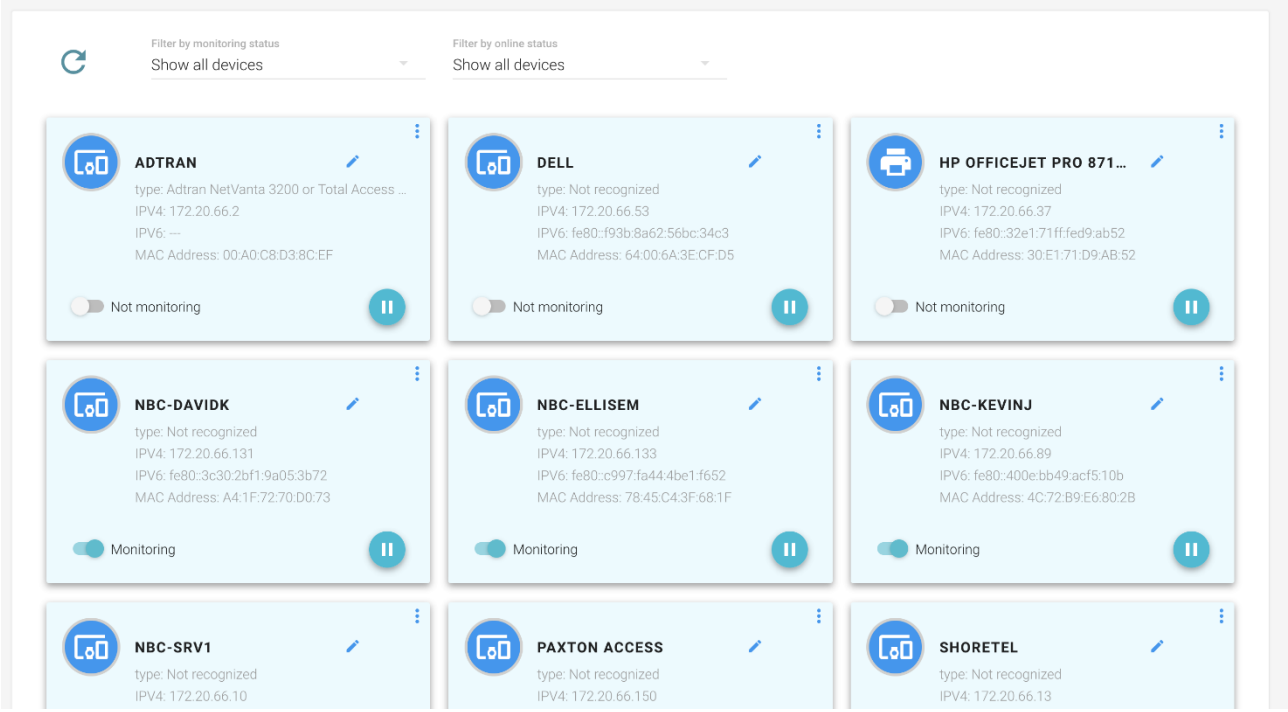


Рисунок 3.15 – Програмно-апаратний комплекс. Список всіх пристроїв

Кожен з пристроїв можна поставити на паузу, тобто натиснувши кнопку, що показана на рисунку 3.16.

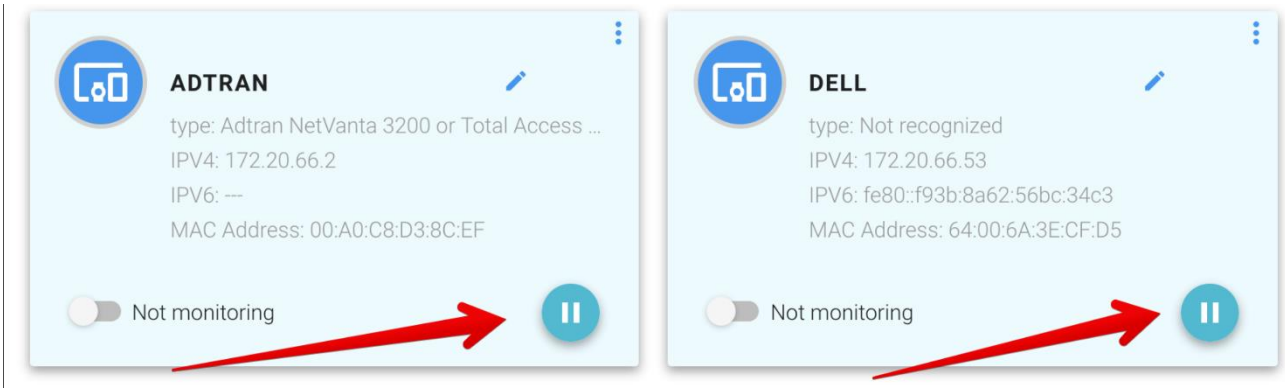


Рисунок 3.16 – Програмно-апаратний комплекс. Блокування доступу до мережі Інтернет для певного пристрою

Натиснувши на перемикач, що знаходіться на карточці пристрою, вмикається та вимикається моніторинг даного пристрою (тобто в режимі реального часу будуть формуватися активності пристрою, проводитись аналіз пакетів на предмет наявності сигнатур, а також поведінковий аналіз), покажемо це на рисунку 3.17.

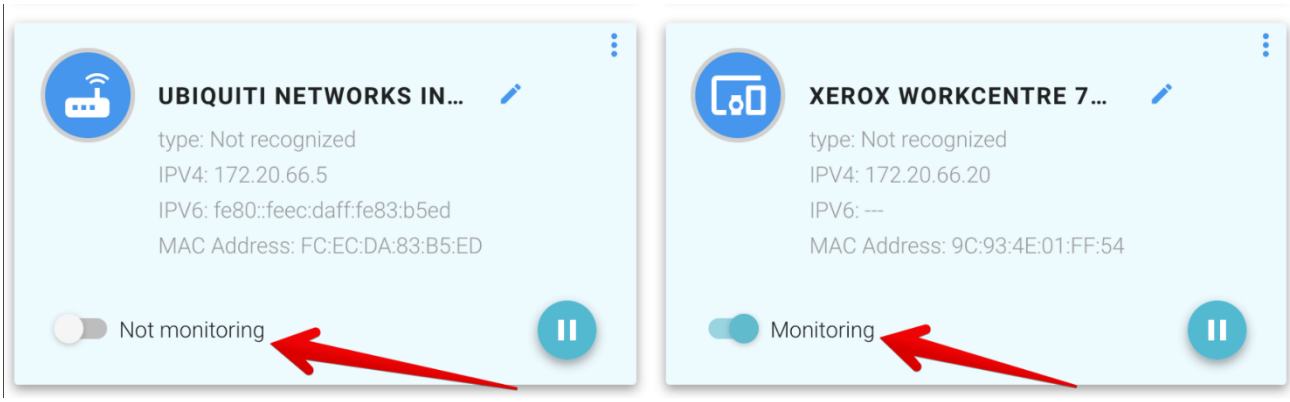


Рисунок 3.17 – Програмно-апаратний комплекс. Вмикання / вимикання захисту та моніторингу активностей певного пристрою.

Перейшовши на сторінку окремого пристрою, видно інформацію по пристрою - назву, IP-адресу, MAC-адресу та час останнього підключення до мережі ,- графік активності за певний день, як зображено на рисунку 3.18

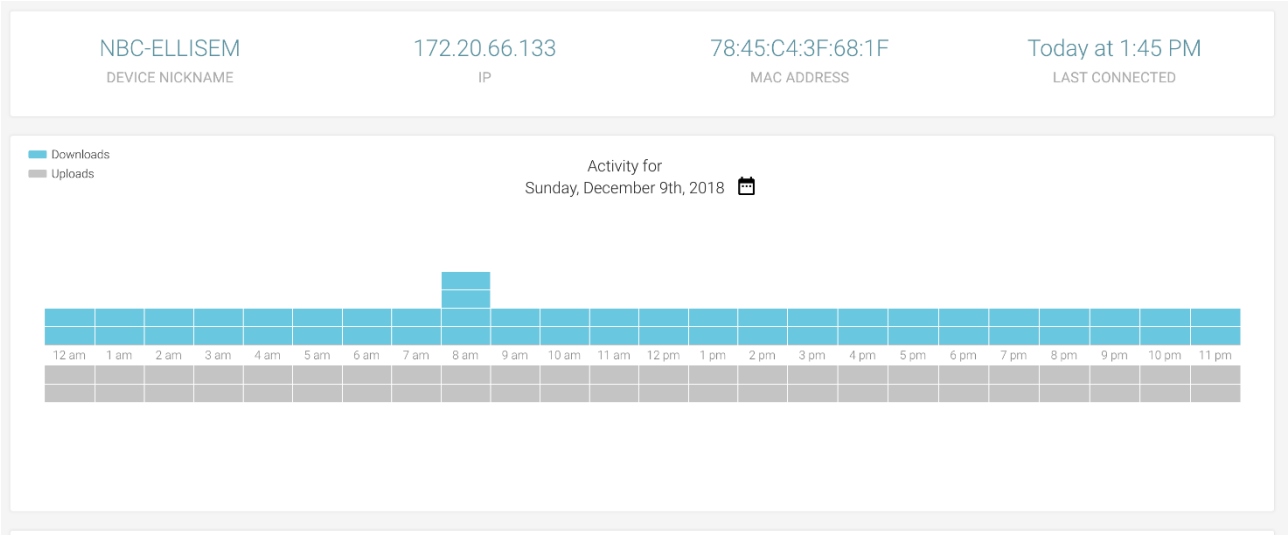
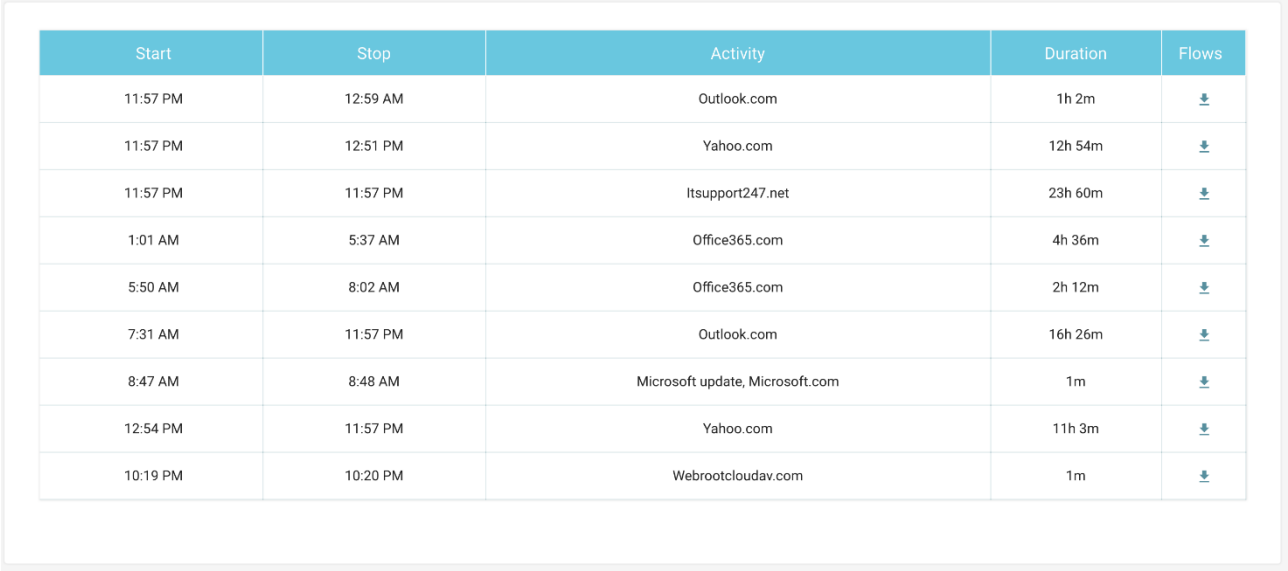


Рисунок 3.18 – Програмно-апаратний комплекс. Графік активності певного пристрою за певну дату.

А також є можливість переглянути звіт активностей за певну дату. Дану можливість представлено на рисунку 3.19



Start	Stop	Activity	Duration	Flows
11:57 PM	12:59 AM	Outlook.com	1h 2m	
11:57 PM	12:51 PM	Yahoo.com	12h 54m	
11:57 PM	11:57 PM	Itsupport247.net	23h 60m	
1:01 AM	5:37 AM	Office365.com	4h 36m	
5:50 AM	8:02 AM	Office365.com	2h 12m	
7:31 AM	11:57 PM	Outlook.com	16h 26m	
8:47 AM	8:48 AM	Microsoft update, Microsoft.com	1m	
12:54 PM	11:57 PM	Yahoo.com	11h 3m	
10:19 PM	10:20 PM	Webrootcloudav.com	1m	

Рисунок 3.19 – Програмно-апаратний комплекс. Звіт активностей певного пристрою за певну дату.

У налаштуваннях доступних користувачу можна вибрати часовий пояс, вказати електронну пошту на яку буде відсилатися щоденний електронний звіт, час в який буде відсилатися щоденний електронний звіт, а також номер версії прошивки, що встановлено на пристрої. Всі ці налаштування можна побачити на рисунку 3.20.

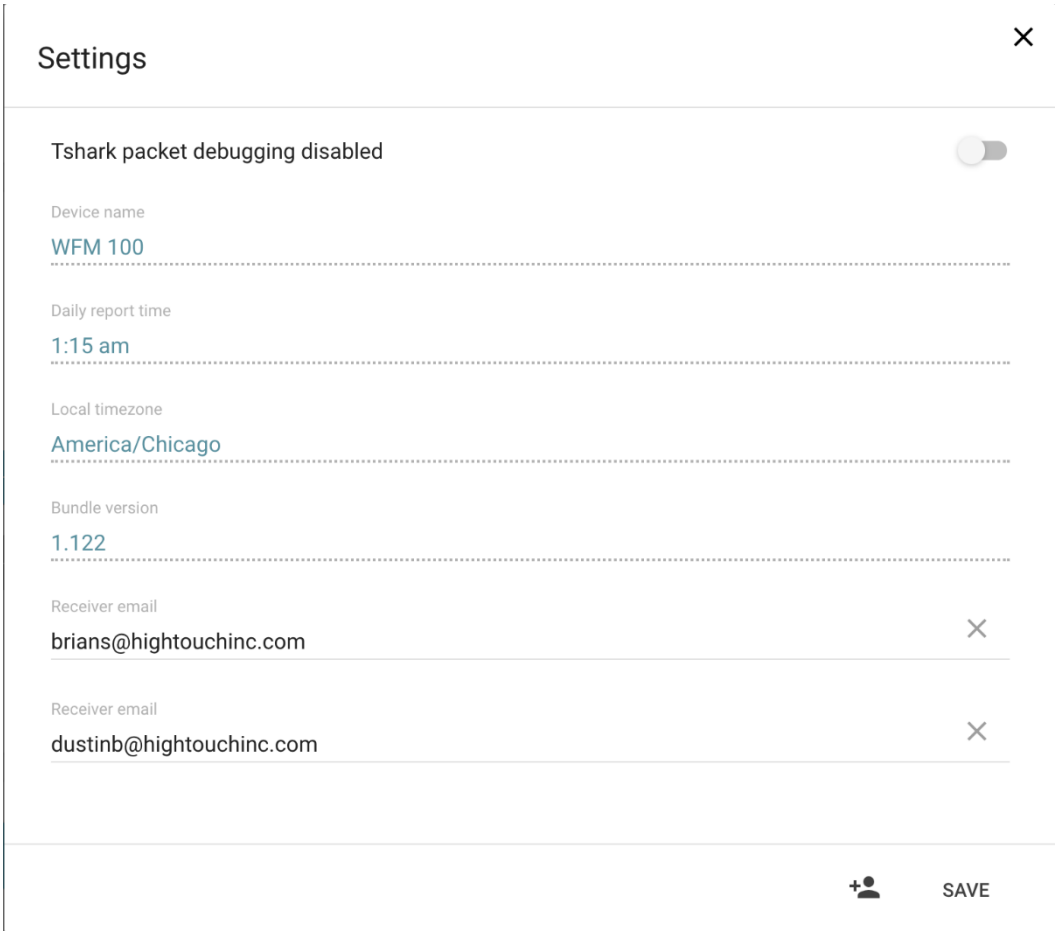


Рисунок 3.20 – Програмно-апаратний комплекс. Налаштування програмно-апаратного комплексу.

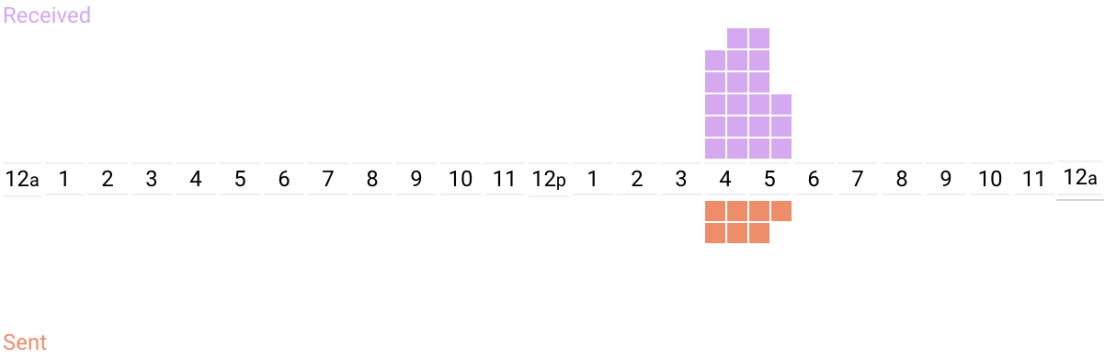
Електронний звіт, що приходить на електронну пошту представлено на рисунках 3.21 та 3.22.

29	29	16	0
total devices	monitored devices	active devices	new devices

Need help identifying your devices? Click here.

See activities that don't make sense? Please let us know by forwarding this message to info@wifimonitor.com

Vinders-MBP



Video 1x (35m)	**Social media** 0x (0m)
Email 3x	**Audio** 0m
Web & other apps 0x	**Messaging** 4x

Рисунок 3.21 – Програмно-апаратний комплекс. Електронний звіт.

Start	End	Activity		Duration
4:50 PM	5:29 PM	**Unknown, Other**	▼	39m
4:52 PM	5:05 PM	Clienthub.us	▼	13m
4:53 PM	4:55 PM	Upwork.com	▲	2m
4:55 PM	4:59 PM	Material.io	▼	4m
4:55 PM	5:30 PM	**YouTube**	▼	35m
4:57 PM	5:03 PM	**Slack**	▼	6m
5:01 PM	5:02 PM	Email	▲	1m
5:06 PM	5:11 PM	**Email**	▼	5m
5:07 PM	5:15 PM	Upwork.com	▲	8m
5:09 PM	5:13 PM	**Slack**	▲	4m
5:16 PM	5:18 PM	Clienthub.us	▼	2m
5:17 PM	5:25 PM	Slack	▲	8m
5:17 PM	5:30 PM	**Email**	▼	13m
5:25 PM	5:25 PM	IMessage	▼	

Рисунок 3.22 – Програмно-апаратний комплекс. Електронний звіт.

Висновок до розділу

У розділі детально описані засоби розробки програмно-апаратного комплексу та наведені основні переваги кожного засобу. Описані вимоги до технічного забезпечення, які потрібні для використання програмно-апаратного комплексу. Вимоги до технічного забезпечення включають вимоги до серверу, на якому буде працювати серверна частина та вимоги до пристрою, на якому буде використовуватись клієнтська частина.

У розділі наведено діаграми класів, послідовності та розгортання, які описують архітектуру системи та її частин, також описані функції та надано інструкцію користувача.

4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

В даному розділі викладено підхід для розробки рішення, яке реалізує викладене в роботі напрацювання. Провівши аналіз ринку були описані цілі та ідея, сформульовані основні вимоги, визначені сильні та слабкі сторони потенційного комерційного продукту в результаті SWOT-аналізу.

4.1 Опис ідеї проекту

Головною метою стартап-проекту є розробка програмно-апаратного комплексу аналізу вразливостей пристроїв однієї підмережі з доступом до мережі Інтернет, який може бути використаний замовниками за призначенням (про це мова піде далі). Розглянемо зміст ідеї, можливі напрямки застосування, основні переваги, які зможе отримати користувач представлено у таблиці 4.1.

Таблиця 4.1 – Переваги для користувача

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмно-апаратний комплекс аналізу вразливостей пристроїв однієї підмережі.	Домашнє використання	Можливість батьківського контролю за дітьми, наявність звітності по активності кожного пристрою домашньої мережі.
	SOHO-ринок	Можливість контролю активності працівників, виявлення атак з мережі Інтернет.
	Enterprise рішення	Можливість контролю активності працівників, виявлення атак з мережі Інтернет.

Продовження таблиці 4.1

	Розумний дім	Контроль IoT пристроїв та профілакика можливості «злити» приватні дані в мережу Інтернет. Передача файлів певних розмірів заздалегідь встановленими адміністраторами підмережі. Це допоможе уникнути витоку даних особистого характеру.
	Персональне використання	Мобільний фаєрвол, який в будь який момент можна підключити до модему по бездротовому з’єднанню

Провівши аналіз ринку було виявлено ряд конкурентів, які представляють свої продукти в даному сегменті. Серед них: Bitdefender Box, CUJO AI, Norton Core, F-Secure SENSE

Bitdefender Box надає функції блокування шкідливих програм, виладення паролів, крадіжок особистих даних та хакерських атак для всіх пристроїв, підключених до Інтернету - навіть тих, у яких немає операційної системи. Дає можливість мережевого та локального захисту для пристроїв Windows, Mac, iOS та Android, вдома та поза межами, а також надає функцію VPN. Дане рішення постачається з високопродуктивним обладнанням для швидкого з'єднання з мережею та практично миттєвою реакцією на всі загрози.

CUJO AI захищає всі пристрої, підключені до маршрутизатора WiFi. Алгоритми AI забезпечують захист від віддаленого доступу, шкідливих програм, фішингу та багато іншого. CUJO AI - це антивірус для всіх бездротових пристроїв вдома. Це частина апаратного забезпечення, призначеного для вдосконалення домашньої безпеки Інтернету. Оскільки багато людей оснащує будинок IoT рішеннями, такими як розумний термостат, смарт-телебачення та

сучасними ігровими консолями, стоїть питання, як можна захистити ці пристрої, що не дозволяють традиційні брандмауери. Після того, як пристрій встановлено, він автоматично виявляє та керує всіма підключеними пристроями одночасно, оберігаючи всі пристрої екосистеми від інтернет-загроз.

Norton Core є бездротовим маршрутизатором, як і багато інших його суперників з двома великими відмінностями: форма його відмінної сфери надає стильному розквіту, незвичному для маршрутизаторів, і має вбудоване програмне забезпечення безпеки для захисту будь-якого пристрою, підключеного до цієї бездротової мережі. Програмне забезпечення буде виявляти та контролювати будь-яке пристрій, підключене до Інтернету, включаючи будь-який пристрій IoT, і буде карантировать його, якщо він почнуть вести себе підозріло. Якщо ви принесете ноутбук, на якому є вірус, Norton Core виявляє це, як тільки пристрій підключено до мережі. Програма відображатиме сповіщення, коли виявляться значні загрози або користувачі мережі намагаються отримати доступ до заблокованих сайтів. На відміну від багатьох інших бездротових маршрутизаторів, Core не виступає як VPN (віртуальна приватна мережа), що вимагає використання додаткового продукту для захисту та анонімності трафіку в мережі, проте мережевий трафік зашифрований.

F-Secure SENSE містить безліч технологій для захисту вашого підключеного дому від загроз.

SENSE виявляє, який пристрій підключено до мережі. Це дозволяє SENSE адаптуватися до своїх потреб безпеки, оскільки різні типи підключених пристроїв вимагають різних типів захисту. Наприклад, загрози, що впливають на ваш домашній ПК, будуть сильно відрізнятися від тих, які впливають на ваш розумний термостат. Безпека IoT виявляє, коли відбувається щось незвичне з розумними пристроями, відстежуючи їх трафік, щоб забезпечити додатковий захист від загроз IoT. Таким чином, SENSE може захистити пристрої без наявності програмного забезпечення безпеки.

4.2 Технологічнй аудит ідеї проекту

Далі наведений аудит технології, за допомогою якої можна реалізувати ідею проекту.

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 4.2):

Таблиця 4.2 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технології
1	Інтерфейс користувача	React.js	Наявні готові бібліотеки з відмінними в плані дизайну елементами	Так
2	Модуль сканування пристроїв підмережі	Node.js	Наявні готові модулі операційної системи (у нашому випадку Nmap)	Так
3	Модуль атаки man-in-the-middle для проходження трафику через EMDS	Node.js	Наявні готові модулі операційної системи (у нашому випадку ARP-spoof)	Так
4	Модуль збору пакетів	Node.js	Наявні готові модулі операційної системи (у нашому випадку Wireshark)	Так

Продовження таблиці 4.2

3	Модуль сигнатурного та поведінкового аналізу	Модель класифікації на основі методу сигнатур та моделювання мереж Петрі	Наявні платні та безкоштовні бібліотеки для створення та використання, а також власні доопрацювання	Так
4	Модуль розпізнавання та інтерпритація пакетів данних в активності	Модель класифікації на основі DNS запитів, DPI протоколів та портів	Розроблена власна логіка для формування активностей	Так
5	Панель адміністрування пристроїв	Node.js, React	Розроблена власна логіка для додавання нових правил розпізнавання активностей, ведення обліку всіх пристроїв, можливість версіонування ПО для пристроїв та реалізація процесу оновлення версії ПО.	Так
Обрана технологія реалізації ідеї проекту: Розробляється програмно-апаратний комплекс у вигляді окремого пристрою на мові Node.js з використанням бібліотеки React для графічної частини, з модулями операційної системи Ubuntu. Акцент у виборі засобів для розробки дизайну впав на готові бібліотеки, що для графічного дизайну – безкоштовні.				

Тож судячи з вище наведеної інформації даний проект можливо реалізувати технічно. Складнощів у використанні та отриманні готових бібліотек виникнути не повинно.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Зробимо аналіз попиту. Далі в таблиці 4.3 наводиться його аналіз, виходячи з наявного ринку.

Таблиця 4.3 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	-
3	Динаміка розвитку ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	-
5	Сертифікація та стандартизація	При встановленні в державні установи, потрібен сертифікат від Державної служби спеціального зв’язку
6	Норма галузевої рентабельності, %	70 %

Ринок збуту вже має своїх фаворитів, проте питання, що порушує продукт є достатньо актуальним на сьогодні, тому сформулювавши правильний маркетинговий план, можна конкурувати з вже відомими брендами.

Визначимо групи потенційних користувачів та сформулюємо набір вимог від кожної групи відносно товару.

В таблиці 4.4 наведено основні характеристики потенційних клієнтів стартап-проекту

Таблиця 4.4 – Основні характеристики потенційних клієнтів стартап-проекту

N п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Наявність звітності за активностями дітей в інтернеті	Батьки / керівництво підприємства	Деякі користувачі хочуть мати розгорнуту статистику по активностям коричтувачів підмережі	- наявність зрозумілого інтерфейсу для користувача - відправлення добових звітів на емейл адресу
2	Можливість обмежувати доступ до мережі Інтернет	Батьки	Деякі користувачі хочуть мати можливість обмежувати доступ окремому пристрою доступ до мережі Інтернет	- наявність можливості обмежувати доступ окремому пристрою підмережі через інтерфейс

Продовження таблиці 4.4

3	Аналіз шкідливого трафіку	Всі групи користувачів	Деякі користувачі хочуть підсилити захистр пристроїв підмережі, встановивши крім локального антивірусу, пристрій, що буде аналізувати трафік, що ще не дійшов до пристрою	- наявність можливості аналізу трафіку на предмет наявності сигнатур - наявність можливості додавання нових сигнатур - наявність можливості аналізу трафіку шляхом поведінкового аналізу
4	Налаштування Black / White lists	Всі групи користувачів	Деякі користувачі хочуть підсилити захист пристроїв підмережі, шляхом повного блокування чи повного дозволу до ресурсу	- можливість налаштування Iptables
5	Аналіз вихідного трафіку IoT пристроїв	Батьки	Деякі користувачі хочуть контролювати щоб приватні дані, що можуть збиратися IoT-пристроями, не були злиті в мережу інтернет	- можливість аналізу вихідного трафіку

Проведемо аналіз ринкового середовища та сформулюємо фактори загроз та фактори можливостей у таблицях 4.5 та 4.6.

Таблиця 4.5 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Поява конкурентів	Можлива поява нових гравців на ринку, що будуть розмивати частку клієнтів	Розробка нових функціональних можливостей продукту, розробка штучного інтелекту для прийняття рішень щодо загроз та віявлення раніше невідомих типів атак
2	Ціна товару	Оскільки процес розробки програмного забезпечення є достатньо затратним, то є можливість	Розробка маркетингового плану, перегляд утворення ціни, рекламування товару

Таблиця 4.6 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Активний піар товару	Продвигання за допомогою якісної реклами	Допомога маркетологів та СММ-спеціалістів
2	Просування товару закордон	Розширення території застосування	Пошук дистрибюторів та іноземних партнерів

Заключним етапом ринкового аналізу, щодо інтеграції продукту є SWOT-аналіз. SWOT-аналіз (або SWOT-матриця) - це метод стратегічного планування, який допомагає людині чи організації визначити сильні сторони, слабкі сторони, можливості та загрози, пов'язані з діловими конкурентами чи плануванням проекту. Даний аналіз має на меті визначити цілі підприємницької діяльності або проекту та визначити внутрішні та зовнішні фактори, які є сприятливими та несприятливими для досягнення цих цілей. Користувачі аналізу SWOT часто запитують та відповідають на запитання, щоб створити значущу інформацію для кожної категорії, щоб інструмент був корисним та визначити їх конкурентну перевагу. SWOT був описаний як спрощений і справжній інструмент стратегічного аналізу.

Назва є абревіатурою для чотирьох параметрів, які розглядає техніка.

Сильні сторони: характеристики бізнесу або проекту, що дають йому перевагу перед іншими.

Слабкі сторони: характеристики бізнесу, що ставлять бізнес або проект у невигідному становищі порівняно з іншими.

Можливості: елементи у навколишньому середовищі, які бізнес або проект можуть використати для його переваг.

Загрози: елементи в середовищі, які можуть спричинити проблеми для бізнесу чи проекту.

SWOT – матриця програмно-апаратного комплексу представлена у таблиці 4.7.

Таблиця 4.7 – SWOT - матриця

Сильні сторони: • висока якість розпізнавання • задоволення потреб ринку • Можливіть переглядати активності пристроїв підмережі • Контроль IoT-пристроїв	Слабкі сторони: • коштовність розробки • ціна товару
Можливості: • розробка нових функціональних можливостей	Загрози: • заповнення ринка новими виробниками • невелика кількість проданих екземплярів

Висновок до розділу

У розділі детально описано аналіз стартап проекту, виконано аналіз ризиків та можливостей, виділено основні групи користувачів та функції, які є привабливими для кожної з груп. Описані вимоги до технічного забезпечення, які потрібні для використання програмно-апаратного комплексу. Вимоги до технічного забезпечення включають вимоги до серверу, на якому буде працювати серверна частина та вимоги до пристрою, на якому буде використовуватись клієнтська частина.

Проект буде захищено від копіювання регістрацією назви програми, створення заявки на отримання патенту на винахід, щоб уберегти алгоритм роботи від копіювання.

ЗАГАЛЬНІ ВИСНОВКИ

Під час виконання магістерської дисертації було досліджено предметну область, визначено вимоги до програмно-апаратного комплексу. Визначено вхідні дані, які надходять та їх джерела. Визначені вихідні дані програмно-апаратного комплексу, приведено їх структуру. Описано та зображено процеси та функції.

На основі даних, отриманих в процесі аналізу була сформульована задача моделювання станів за допомогою мереж Петрі.

В результаті проведеного дослідження було розроблено програмно-апаратний комплекс аналізу вразливостей, що використовує альтернативний архітектурний підхід для впровадження в підмережу, шляхом під'єднання до роутеру та проведення атаки man-in-the-middle на решту пристроїв підмережі. Тобто фізичної взаємодії програмно-апаратного комплексу і пристроїв підмережі немає, проте програмно-апаратний комплекс виступає в якості файєрволу в для всіх дейвайсів підмережі.

Наразі програмно-апаратний комплекс вирішує наступні кейси:

- блокування відомої атаки. Напрямлена атака проводиться на один з мережевих пристроїв підмережі. У потоці пакетів, що пройшли через EMDS, одна з двох фаз – сигнатурний чи поведінковий аналіз - виявила загрозу. Пакет, в якому було виявлено атаку, і наступні пакети відповіді заблоковані. Таким чином, відповідь не буде повністю інтерпретована на кінцевому пристрої, і атака не буде виконана;
- налаштування black / white листів. Є можливість формування списків доступу до електронних ресурсів – список завжди дозволених ресурсів та список заборонених ресурсів;
- формування звіту активності кожного пристрою підмережі.

Наведена інструкція користувача по експлуатації програмно-апаратного комплексу. Описані кроки необхідні для виконання всіх функцій веб застосунку, наведені копії екранних форм. Описані випробування програмного продукту на відповідність функціональним вимогам.

ПЕРЕЛІК ПОСИЛАНЬ

1. Сервер - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80.
2. ReactJS - значення слова [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/ReactJS
3. Моделювання та аналіз безпеки розподілених інформаційних систем : М74 навч. посіб. [для студ. спец. 121 «Інженерія програмного забезпечення»] / В. В. Литвинов, В. В. Казимир, І. В. Стеценко та ін. – Чернігів : Чернігів. нац. технол. ун-т, 2016. – 254 с.
4. ECMAScript - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/ECMAScript.
5. HTML - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/HTML.
6. HTTP - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/HTTP.
7. CSS - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/CSS.
8. Node.js - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Node.js.
9. MongoDB - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/MongoDB.
10. ГОСТ 19781-90 Термины и определения. [Електронний ресурс] - Режим доступу до ресурсу: http://www.rugost.com/ .
11. Петрі - об’єктна модель поширення кібератак в РІС. Стеценко І.В. – С 4. [Електроний ресурс], режим доступу - http://simulation.su/uploads/files/default/2018-litvinov-stecenko.pdf.
12. Литвинов В.В. Петрі-об’єктна модель системи управління розподіленими обчислювальними ресурсами / В.В. Литвинов, І.В. Стеценко // Тези доповідей Міжнародної науково-практичної

конференції „Інформаційні технології в освіті, науці і техніці”(ІТОНТ2012): Черкаси, 25-27 квітня 2012р. – У 2 т. – Черкаси: ЧДТУ, 2012. – Т.1. – С.33-34.

13. Стеценко И.В. Алгоритм імітації Петрі-об’єктної моделі / І.В. Стеценко // Математичні машини і системи. – Київ, 2012. - №2 . №1 . – С.154-165.
14. Белова А. Л., Бородавкин Д. А. Сравнительный анализ систем обнаружения вторжений [Электронный ресурс] / А. Л. Белова, Д. А. Бородавкин. – Режим доступа: https://cyberleninka.ru/article/n/sravnitelnyy-analiz-sistem-obnaruzheniya-vtorzheniy
15. Смыков Г. Новый взгляд на обнаружение и предотвращение web-атак [Электронный ресурс] / Г. Смыков. – Режим доступа: https://www.securitylab.ru/contest/290792.php
16. Войтович О. П., Ювковецький О. С. Класифікація вразливостей Web-ресурсів [Електронний ресурс] / О. П. Войтович, О. С. Ювковецький. – Режим доступу: http://itce.pu.if.ua/files/topics/Voytovych-Yuvkovetskyi.pdf

ДОДАТОК А Графічний матеріал

Структурна схема бази даних

Схема спілкування пристроїв підмережі після впровадження програмно-апаратного комплексу

Структурна схема діяльності визначення типу атаки

Структурна схема послідовності визначення типу атаки

Структура даних окремого мережевого пакету

Копія екранної форми «Приклад звіту активностей пристрою»

Копія екранної форми «Створення правила для розпізнавання активностей»